

Cecile Simon-Bellamy

TIETOMALLINNUSOHJELMISTON KE- HITTÄMINEN SUUNNITTELUTOIMISTON TARPEISIIN AVOINTA OHJELMISTORA- JAPINTAA KÄYTTÄEN

Rakennetun ympäristön tiedekunta
Diplomityö
Kesäkuu 2019

TIIVISTELMÄ

Cecile Simon-Bellamy: Tietomallinnusohjelmiston kehittäminen suunnittelutoimiston tarpeisiin avointa ohjelmistorajapintaa käyttäen
Diplomityö
Tampereen yliopisto
Rakennustekniikka
Kesäkuu 2019

Tämän diplomityön tarkoituksena on tutkia, kuinka tietomallinnusohjelmistoa kehitetään suunnittelutoimiston tarpeisiin paremmin sopivaksi. Asiaa tutkitaan vain kohdeyrityksen näkökulmasta ja tietomallinnusohjelmistona tutkitaan Trimblen tarjoamaa tietomalliohjelmisto Tekla Structuresia ja sen avointa rajapintaa Tekla Open API:a.

Avoimiin rajapintoihin tutustutaan ensin yleisesti kirjallisuustutkimuksen avulla ja lisäksi perehdytään muutamaan laajempaan avoimien rajapintojen kokonaisuuteen. Avoimilla rajapinnoilla on tärkeä merkitys modernissa ohjelmistokehityksessä ja innovaatioiden synnyssä.

Tekla Open API:n tarjoamia mahdollisuuksia ovat laajennukset, joiden kolme päätyyppiä ovat applikaatiot, pluginit ja custom propertyt. Avoimen rajapinnan kautta Tekla Structures ohjelmisto on mahdollista muokata vastaamaan työntekijöiden ja yrityksen tarpeita.

Yrityksen ja käyttäjien tarpeita tutkitaan keskustelemalla ensin kohdeyrityksen BIM managerin kanssa ja sen jälkeen toteuttamalla kyselytutkimus ohjelmiston käyttäjille. Pyrkimyksenä on selvittää, mikä ohjelmistossa koetaan työläimpänä ja mihin kaivataan ratkaisuja. Ohjelmistokehitystä tehtäessä on muistettava, että kehitystyötä ja työkaluja tehdään käyttäjille eikä kehittäjille ja se pyritään varmistamaan kysymällä laajasti käyttäjien näkemyksiä ja mielipiteitä.

Selvityksen perusteella kehitystyölle on kysyntää ja se koetaan mahdollisuutena. Kehitystyötä suunniteltaessa ja resursseja mietittäessä on huomioitava, että vaikka Tekla Open API:a mainostetaan helppona tapana kehittää omia toiminnallisuuksia ja työkaluja, vaatii se jonkin verran osaamista ohjelmoinnista ja mikäli työkaluja halutaan kehittää ja ylläpitää, vaatii se osaavat tekijät. Käyttäjät on syytä osallistaa jatkuvasti mukaan kehitystyöhön aina perehdytyksestä alkaen.

Parametrinen mallintaminen nousee käyttäjien vastausten perusteella kiinnostavaksi teemaksi ja sen mukaan ottamista avoimen rajapinnan kehitystyöhön on syytä harkita, parametrisella mallintamisella voidaan pureutua samoihin ongelmiin kuin avoimen rajapinnan työkaluilla, eli tehostamaan ja automatisoimaan toistuvia suunnittelutehtäviä ja vapauttamaan suunnittelijoiden työaika muihin tehtäviin.

Avainsanat: Tekla Open API, avoin rajapinta, parametrinen mallinnus.

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

Cecile Simon-Bellamy: Developing building information modeling software with open application programming interface
Master of Science Thesis
Tampere University
Master's Degree Programme in Civil Engineering
June 2019

The aim of this thesis is to explore how to develop building information modeling software to meet the needs of structural designers using open application programming interface. The scope of this thesis contains only the perspective for the target company and the only applied API is Tekla Structures Open API provided by Trimble.

Open application interfaces are studied with the help of literature research and in addition a broader set of APIs is introduced. APIs play an important role in modern software innovation and creating innovations.

The possibilities offered by the Tekla Open API can be divided in three main types: applications, plugins and custom properties. By using and taking the advantage of the Tekla Open API the Tekla Structures software can be customized to meet the needs of employees and company more efficiently.

The needs of the company and the users are studied by discussing with the company's BIM manager and then conducting a survey for the users. The aim is to find out and where solutions are needed. When developing a software, one must remember that everything is done for the users, not for the developers. This is ensured by gathering broad views and opinions from the users.

Based on the survey, there is great demand for development work, and it is perceived as an opportunity. When thinking about existing resources, it should be noted that even though Tekla Open API is known as an easy way to develop tools and functionalities, it requires at least basic skills in programming. Users should be constantly involved in the development work, starting with the orientation process.

Parametric modeling rises as an interesting theme based on user responses. It should be considered a one aspect when developing modeling since it is an alternative solution for automating recurring tasks.

Keywords: Tekla Open API, open application programming interface, parametric modeling

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Kiitokset Pöyry Finland Oy:lle tämän diplomityön mahdollistamisesta ja rahoittamisesta. Haluan erityisesti kiittää ohjaajani Kaarle Koskelaa Pöyry Finland Oy:stä, loputon ymmärtäväisyytesi ja kärsivällisyytesi osoittautui kultaakin kalliimmaksi. Kiitos myös muille Pöyryn puolesta tähän työhön matkan varrella osallistuneille. Tampereen yliopiston professori Mikko Malaskalle ja Kristo Melalle kiitokset työni tarkastamisesta ja kaikista kommenteista.

Kiitos perheelleni jatkuvasta uskosta ja tuesta opiskeluideni aikana. Kiitos TTY kaikista uskomattomista mahdollisuuksista ja huikeista seikkailuista. Kiitos opiskeluvuosieni perheelle, maailman upeimmalle kerholle. Te opetitte minulle enemmän kuin kaikki 300 opintopistettäni yhteensä. Kiitos kun sain olla osa tarinaa.

Toisinaan matka on määräävää tärkeämpi, niin oli myös tämän diplomityön kanssa. Viimeisenä haluan kiittää ihmisiä, jotka tekivät matkan loppuun saattamisesta mahdollista. Kiitos Ella jokaisesta alitetusta rimasta. En koskaan olisi uskonut kuinka syvälle olemme valmiit vajoamaan. Kiitos kaikki AKSUn Nemoset jatkuvasta hiillostuksesta ja painostuksesta, sitä tarvittiin. Viimeisimpänä kiitos Johannes. Ilman sinua tämä työ ei olisi koskaan päätynyt kansiin.

Tampereella 19.6.2019,

Cecile Simon-Bellamy

SISÄLLYSLUETTELO

1. JOHDANTO	1
1.1 Tutkimuksen tausta	1
1.2 Tutkimuksen tavoite ja tutkimuskysymykset	1
1.3 Työn rajaukset	2
1.4 Työn rakenne ja tutkimusmenetelmät	2
2. OPEN API:T	4
2.1 Open API	4
2.1.1 Historia	4
2.1.2 Avointen rajapintojen merkitys	5
2.1.3 Määritelmät	8
2.2 Esimerkkejä käytännön sovellutuksista	9
2.2.1 6Aika	10
2.2.2 Pankkien avoimet rajapinnat	11
2.2.3 Paikkatiedot	12
3. TEKLA STRUCTURES OHJELMISTO	13
3.1 Tekla Open API	13
3.1.1 Ohjelmistotuotanto	13
3.1.2 Käyttöympäristö	14
3.2 Open API-toiminnot	17
3.2.1 Applikaatiot	18
3.2.2 Pluginit	18
3.2.3 Custom property	20
3.3 Uusimmat työkalut ja kehitteillä olevat mahdollisuudet	20
3.3.1 Tekla Tedds	20
3.3.2 Tekla Developer Center	21
4. SELVITYS OPEN API-SOVELLUSTEN KÄYTÖSTÄ KOHDEYRITYKSESSÄ	22
4.1 Toiminta-alueet, joilla kohdeyritys hyödyntää Tekla Structures - ohjelmistoa	22
4.2 Tavoitteiden selvitys	22
4.2.1 Keskustelu BIM managerin kanssa	23
4.2.2 Kyselytutkimus käyttäjille	23
4.2.3 Kyselyn tulokset	25
4.2.4 Tulosten analyysi	26
4.3 Avointen rajapintojen käyttöön liittyvät keskeiset haasteet ja mahdollisuudet	27
5. OPEN API -SOVELLUSTEN KÄYTÖN LAAJENTAMINEN JA TEHOSTAMINEN KOHDEYRITYKSESSÄ	29
5.1 Kehitystyössä huomioitavaa	29
5.1.1 Perehdytys	30
5.1.2 Jatkuva palaute ja käyttäjien osallistaminen	32
5.1.3 Avoimen rajapinnan sisäinen avoimuus	34

5.2	Piirustukset	34
5.3	Parametrinen mallintaminen.....	35
5.3.1	Grasshopper	36
5.3.2	Tekla live link	36
5.3.3	Tulevaisuuden näkymät	37
6.	YHTEENVETO.....	39
	LÄHTEET	41
	LIITE A: KYSELYTUTKIMUKSEN KYSYMYKSET	44

LYHENTEET JA MERKINNÄT

.dll	Dynamic-link library, tiedostopääte
.exe	Executable, tiedostopääte
API	Application Programmin Interface, avoin ohjelmointi rajapinta
BIM	Building Information Model, tietomalli
C#	C Sharp, oliopohjainen ohjelmointikieli
C++	Alemman tason ohjelmointikieli
CLR	Common Language runtime, ajoympäristö
CLS	Common Language Spesification
EU	Euroopan Unioni
FEM	Finite Element Method, elementtimenetelmä
IDE	Integral Development Environment, kehitysympäristö
IL	Intermediate Language
JIT	Just-In-Time, kääntäjä
NURBS	Non-Uniform Rational Basis Spline, matemaattinen malli pintojen ja käyrien luomiseen ja esittämiseen
WFS	Web Feature Service, määrittelee rajapinnan paikkatietokohteiden etsintään, kyselyyn ja muokaamiseen tietoverkon yli.
WMS	Web Map Service, rajapinta paikkatiedon muodostaman kartan lataamiseen kuvina tietoverkon yli.

1. JOHDANTO

Tässä työssä tutkitaan, kuinka tietomallinnusohjelmistoa kehitetään parhaiten vastaamaan suunnittelutoimiston tarpeita avointa ohjelmistorajapintaa käyttäen. Kohdeyritys on tässä työssä Pöyry Finland Oy ja tutkittava tietomallinnusohjelmisto Tekla Structures.

1.1 Tutkimuksen tausta

Rakennesuunnittelijat käyttävät kohdeyrityksessä pääasiallisena tietomallinnusohjelmistona Tekla Structures-ohjelmistoa. Ohjelmisto on tällä hetkellä hyvin kirjavassa käytössä erilaisissa suunnittelu- ja konsulttiyrityksissä ympäri maailmaa ja se tehdään laajalle käyttäjäryhmälle. Kohdeyritys ei tee perinteistä talonrakennuksen rakennesuunnittelua vaan keskittyy teollisuuden tarpeisiin. Projektit ovat usein hyvin laajoja ja monimutkaisia eikä Tekla Structures sellaisenaan vastaa kaikkiin kohdeyrityksen tarpeisiin. Tekla Structuresille on olemassa avoin ohjelmointirajapinta Tekla Open API (Application Programming Interface), tai lyhennettynä .NET API, jolla ohjelmistoon on mahdollista rakentaa lisäosia, jotka helpottavat ja tehostavat suunnittelijan työtä. Tekla Structures ohjelmisto kaupallisessa muodossaan on vain puolivalmis kohdeyrityksen tarpeisiin.

Kohdeyritykselläkin on tehty omia sovelluksia Tekla Open API:a hyödyntäen, mutta nyt kehitystyötä halutaan entisestään tehostaa ja laajentaa. Uusien ratkaisujen kehittämisen ohella vanhoja sovelluksia päivitetään ja ylläpidetään mahdollisimman tehokkaasti ja päivitetään vastaamaan uusia ohjelmistojulkaisuja.

Teklan avoimella rajapinnalla on mahdollista kohtuullisen helposti tehdä lisäosia, jotka esimerkiksi automatisoivat tiettyjä toimintoja ja helpottavat suunnittelijoiden työtä. Jos pienellä panostuksella helpotetaan suunnittelijoiden työtä, tarkoittaa se myös suunnittelun tehostumista. Tietomalleja käytetään yhä enemmän ja esimerkiksi tiedonhakua malista voidaan helpottaa suuresti rajapinnan avulla tehdyillä sovelluksilla.

1.2 Tutkimuksen tavoite ja tutkimuskysymykset

Tutkimuksen tavoitteena on kartoittaa, miten voimme hyödyntää rajapintaa kohdeyrityksen tarpeisiin rakennesuunnittelun toimialalla. Kehitystyö ei saa olla itseisarvo vaan sillä pitää saavuttaa hyötyä. Tutkimuksen lopputuloksena on esitys siitä, että mitä kaikkea

kohdeyrityksen kannattaisi lähteä Tekla Structuresin avoimen rajapinnan avulla kehittämään ja tekemään ja kuinka kehitystyöhön kannattaa lähteä. Tavoitteena on myös tarjota näkemystä siitä, kuinka kehitystyö ohjataan jatkossakin suurimpiin kipukohtiin ja varmistetaan, että rivisuunnittelijat hyötyvät uusista työkaluista.

1.3 Työn rajaukset

Aihetta käsitellään vain kohdeyrityksen, Pöyry Finland Oy:n, näkökulmasta. Tietomallinnusohjelmistoista tässä diplomityössä keskitytään tarkastelemaan vain Tekla Structures -ohjelmistoa. Ohjelmiston käyttäjiksi on tässä työssä rajattu vain rakennesuunnittelijat, vaikka käyttäjiä löytyy muiltakin liiketoiminnan osa-alueilta.

Tarkoituksena on selvittää, miten hyödyntää rajapintaa, jotta saadaan tehokkaammin ohjelmisto käyttöön ja samalla suunnittelu tehostuu. Keskitytään siis selkeästi suunnittelun kannalta olennaisiin asioihin, jotka on mahdollista toteuttaa kohtuullisella panostuksella ja olemassa olevilla resursseilla. Työn tarkoituksena ei ole tuottaa valmiita työkaluja vaan tarjota suuntaviivoja kehitystyön laajentamiselle. Työssä ei perehdytä niinkään kohdeyrityksen historiaan avointen rajapintojen kanssa vaan kehitystyön laajentamiseen suhtautaan ennakkoluulottomasti ja avarakatseisesti.

1.4 Työn rakenne ja tutkimusmenetelmät

Ohjelmistorajapinnan hyödyntämistä tutkitaan käyttäen haastatteluita ja käyttäjäkyselyä yrityksen sisällä, ohjeita ja nettifoorumeita, jotka on suunnattu juuri ohjelmistorajapinnan kehittäjille, sekä omaa empiiristä kokemusta. Lisäksi avoimiin rajapintoihin ja sen tarjoamiin sovelluksiin tutustutaan kirjallisuuskatsauksella.

Työn toisessa luvussa käsitellään avoimia rajapintoja yleisesti, niiden historiaa, kehitystä ja merkitystä ohjelmistokehitykselle ja yrityksille. Käydään läpi avoimen rajapinnan määritelmä ja esimerkkejä käytännön sovellutuksista.

Kolmannessa luvussa tarkastellaan ensin Tekla Structuresia tietomallinnusohjelmistona ja sen käyttöympäristöä. Tämän jälkeen kuvaillaan Tekla Structuresin avoimen rajapinnan erilaisia hyödyntämismahdollisuuksia applikaatioiden, pluginien ja custom propertyjen muodossa.

Neljäs luku on selvitys avointen rajapintojen käytöstä yrityksessä. Luvussa kerrotaan valituista tarkastelutavoista ja avataan haastattelun ja käyttäjäkyselyn tuloksia. Lopuksi analysoidaan kohdeyrityksen tarpeet tulevaisuuden kehitystyölle.

Viidennessä luvussa annetaan kehitysideoita ja suuntaviivoja kehitystyön laajentamisen tueksi, sekä teknisiä ajatuksia, että henkilöstöjohtamiseen ja työtyytyväisyyteen liittyviä kysymyksiä.

2. OPEN API:T

API (Application Programming Interface) on lyhenne englanninkielisistä sanoista application programming interface, joka tarkoittaa sovellusohjelmointirajapintaa, joka yleensä lyhennetään rajapinnaksi. Rajapinta on joukko standardisoituja komentoja, joiden avulla tietokoneohjelmat voivat kommunikoida keskenään. Ohjelmistokehittäjät voivat rajapintojen avulla tuottaa ja kehittää erilaisia sovelluksia. (Butterfield & Gerard, 2018)

Open API eli avoin rajapinta määritellään rajapinnaksi, jonka kaikki ominaisuudet ovat julkisia ja kaikkien käytettävissä ehdoilla. Rajapintaa hyödyntävä ohjelma voidaan laatia ilman erillisiä lisenssimaksuja tai omistajan hyväksyntää. Rajapinnan kuvaus ja sen dokumentaatio on julkisesti saatavilla ja rajapinta avoimesti ja vapaasti käytettävissä sovellusten tekemiseksi ja testaamiseksi, jotta avoimuus toteutuu. (Poikola et al., 2014)

Tässä luvussa käsitellään lyhyesti Open API:en historiaa ja käyttötarkoituksia, erilaisia sovelluksia yleisesti ja tarkemmin konsulttitoimistojen näkökulmasta. Lisäksi kappaleen luvussa kerrotaan tarkemmin kolmesta merkittävästä avoimen rajapinnan sovelluksesta.

2.1 Open API

Tässä alaluvussa käsitellään Open API:n eli avoimen rajapinnan määritelmä ja erilaiset standardit, avoimien ja suljettujen rajapintojen historiaa ja rajapintojen merkitystä ohjelmistokehitykselle ja yrityksille.

2.1.1 Historia

Suuri ongelma ohjelmistokehityksessä menneisyydessä oli, että kehittäjille annettiin hyvin tiukat rajaukset ja käyttöliittymät. Tämä yhdistettynä tiukkaan turvallisuuteen, raskaaseen byrokratiaan ja jopa järjettömiin rajauksiin, ei jättänyt tilaa mielikuvitukselle tai uudenlaisille ajattelumalleille. Lopputuloksena oli raskaita ja vaikeasti muokattavia ratkaisuja, jotka tarpeiden muuttuessa oli tehtävä kokonaan uudestaan. Tähän ratkaisuksi kehitettiin rajapinnat. (Thuray, 2013)

Rajapinnat ovat usein hyvin yksinkertaisia, niiden tarkoitus on jakaa pääsy omaisuuteen, jota voivat olla data, prosessit, systeemit tai muu vastaava. Kun tieto jaetaan rajattomalle joukolle sovelluskehittäjiä, tarjotaan tilaisuus rakentaa jotain uutta ja erilaista, ja luoda lisää arvoa yritykselle. Tämä kannustaa kehittäjiä luomaan toinen toistaan parempia ratkaisuja. Kaikkea yrityksen dataa ei välttämättä kannata jakaa avoimesti vaan toisinaan on syytä luoda sisäinen rajapinta ja avoin rajapinta, joissa jaetaan joko kokonaan eri

asioita tai jaettavan datan laajuus vaihtelee. Tällä hetkellä trendinä on, että yrityksen sisäiset kehittäjät työskentelevät yhdessä ulkopuolisten kehittäjien kanssa ja esimerkiksi lisäävät ulkopuolisten kehittäjien tuotoksiin oman osaamisensa yrityksen vain sisäisesti jaetuilla materiaaleilla. Jakamalla rajapinnat eri kategorioihin voi yritys hyödyntää avoimuuden tarjoamaa mahdollisuutta uusiin näkemyksiin ja ideoihin, samalla suojellen ydinosaamista ja liiketoimintaa. Tämä järjestely hyödyttää kaikkia osapuolia. (Thuray, 2013)

Rajapintojen synnystä ei osata sanoa tarkasti, mutta varhaisimpia rajapintoja voidaan pitää hotellien varausjärjestelmiä ja postimyyntiluetteloita. Rajapinnat luotiin alkujaan tietokoneiden keskinäiseen kommunikointiin, eli silloiksi eri ohjelmistojen välille. Rajapintojen avaaminen sovelluskehittäjien käyttöön tapahtui alun perin kehittäjien toiveesta, eräänä hyvä esimerkkinä voidaan käyttää googlen karttapalveluita. Googlen karttojen päälle on rakennettu lukemattomia sovelluksia. Rajapintojen synnyssä ja kehityksessä ratkaisevassa asemassa on ollut yrittäjähenkisyys ja pohdinnat, kuinka hyödyntää taustalla pyöriviä palveluita uudella ja kannattavalla tavalla. (Cater, 2013)

Rajapinnoilla siinä muodossa kuin me tunnemme ne nykyään, on kohtuullisen lyhyt historia. Paine avointen rajapintojen kehittämiseen tuli pääasiassa yritysten ulkopuolelta eikä suinkaan yritysten sisältä. Tarve rajapinnoille on kasvanut, ihmiset vaativat informaation avoimuutta hallituksilta ja sosiaalisen median paine on odotettavissa, mikäli yritys ei luo avointa rajapintaa. Rajapintojen avaaminen lisää myös mahdollisuuksia innovaatioille. Rajapinnat sekoittavat teknologiaa, liiketaloutta ja politiikkaa uudella tapaa, joka on läpinäkyvä, itsepalvelu periaatteella toimiva ja lisää innovaatioita. (Cater, 2013)

Avoimet rajapinnat tekevät tulevaisuudessa ohjelmistokehityksestä entistä ketterämpää. Kun sovellus on rajapinnan kautta avattu muidenkin kehitettäväksi, ketteryys tehostuu ja voidaan kehittää monta eri ominaisuutta samanaikaisesti. Mikropalveluarkkitehtuurit ja avoimet rajapinnat ovat tulevaisuutta. Mikropalveluarkkitehtuurit ovat ohjelmistokehitysmenetelmä, jonka tarkoitus on tehdä sovelluksia helposti käyttöön otettaviksi yksittäisiksi komponenteiksi, jossa yksi sovellus toteuttaa vain yhtä tehtävää. Sovellukset toimivat itsenäisesti, mutta kommunikoivat keskenään, jolloin toimintoja voi yhdistellä. Näiden ansiosta kehittäjillä on vapautta käyttää rajapintojen integraatiota innovaatioihin. (Zumeran, 2017)

2.1.2 Avointen rajapintojen merkitys

Rajapintojen merkitys on ollut erityisen suuri verkkokaupoissa, matkapuhelin sovelluksissa, sosiaalisessa mediassa ja pilvipalveluissa. (Zumeran, 2017) Suurelle yleisölle tutuimpia Open API sovellusten tarjoajia ovat sosiaalisessa mediassa Facebook, yksi suu-

rimmista verkkokaupoista eli Amazon ja lukuisten eri karttapalveluiden pohjanakin toimiva Google Maps. Nämä yritykset lukuisten muiden yritysten tavoin tarjoavat jatkuvasti dataansa kolmansien osapuolien käyttöön, mahdollistaen uusien järjestelmien rakentamisen alkuperäisten päälle tai alkuperäisen sisällön julkaisun uudella alustalla. (Aitamurto & Lewis, 2013)

Intelin teknologiajohtaja Andy Thurai käsittelee artikkelissaan ”How APIs Fuel Innovation” rajapintojen merkitystä innovaatioiden synnyssä. Vaikka artikkeli käsittelee rajapintoja yleisesti, voidaan ajatuksia soveltaa erityisesti myös avoimille rajapinnoille. Artikkelin mukaan rajapinnat avaavat uusia tulonlähteitä, luovat uusia kumppanuuksia, helpottavat liikekumppaneiden kommunikointia ja erityisesti luovat nopeamman kanavan innovaatioille. Ennen rajapintojen yleistymistä kehitystyö oli hitaampaa ja erityisesti yritysten ja sovelluskehittäjien kommunikointi vaikeampaa. Rajapintojen avulla ohjelmistokehittäjien on mahdollista testata ja tutkia ohjelmiston toimintaa jatkuvasti koko kehitysvaiheen ajan ja havaita ongelmat ja mahdollisuudet ajoissa. (Thuray, 2013)

Blogijulkaisussaan Patrick Emmons listaa neljä avointen rajapintojen tuottamaa etua asiakkaalle, kumppanille ja palvelun tarjoajalle. Julkaisussaan Emmons korostaa erityisesti sitä, että kaikki osapuolet voivat parhaimmillaan hyötyä avoimesta rajapinnasta. Hyödyiksi hän listaa jo aiemminkin mainitut rajojen poistamisen asiakkaan ja ohjelmistokehittäjän sekä palvelun tarjoajien väliltä sekä innovaatioiden edistämisen, mutta mainitsee myös asiakkaan näkemysten hyödyntämisen ja organisaation oman toiminnan tehostamisen. (Emmons, 2016)

Avoimet rajapinnat voivat käyttäjien ja asiakkaiden lisäksi hyödyttää myös palvelun tarjoaja organisaation henkilöstöä ja tehostaa organisaation toimintaa. Samaa rajapintaa hyödyntäen esimerkiksi palvelupyynnöt voidaan ohjata suoraan vapaille resursseille tai yhteistyökumppaneille. Pyyntöjen ja palveluiden automatisointi tehostaa organisaation sisäistä toimintaa parantaen samalla asiakkaiden kokemusta. (Emmons, 2016)

Datan kerääminen käyttäjien toiminnasta ei ole uutta. Kun halutaan ymmärtää miten loppukäyttäjät hyödyntävät sovelluksia, kerätään tietoa vuorovaikutuksesta käyttäjien ja sovelluksen välillä. Tätä dataa analysoidaan usein käyttäjien näkökulmasta, sen sijaan, että tutkitaan miten sovellus tai palvelu hyödyttää koko loppukäyttäjän organisaatiota. Kun integroidaan palvelu tai sovellus osaksi loppukäyttäjän, eli asiakas organisaation omia sovelluksia, voidaan helpommin tarkkailla palvelun käytettävyyden lisäksi sitä, miten ja mihin palvelua hyödynnetään. Tämän datan avulla palvelua voidaan kehittää yhä

paremmin vastaamaan asiakkaiden tarpeita. Tyytyväinen asiakas on vähemmän halukas vaihtamaan käyttämäänsä palvelua ja hyvällä asiakaspalvelulla voidaan vaalia pitkiä asiakassuhteita. (Emmons, 2016)

Tanja Aitomurron ja Seth C. Lewisin artikkeli ”Open innovation in digital journalism: Examining the impact of Open APIs at four news organizations” tutkii avoimien rajapintojen tuottamaa lisäarvoa neljälle suurelle yhdysvaltalaiselle uutistoimistolle. Tutkimuksessa havaittiin, että uutistoimistot, jotka jakavat uutisaineistojaan avoimen rajapinnan kautta saivat näkyvyyttä uutisilleen laajalti ympäri internetiä, joka tekee uutistoimiston tavaramerkkiä näkyvämmäksi ja tunnetummaksi. Avoimien rajapintojen käyttöönotto on ensimmäinen askel kohti avoimia innovaatiota digitaalisessa mediassa. Avoimien rajapintojen avulla organisaatiot tehostivat innovaatioprosessejaan organisaation sisällä, mutta myös organisaatorajojen ulkopuolella. (Aitamurto & Lewis, 2013)

Uutistoimistot kilpailevat artikkelin mukaan paitsi nopeudessa, myös kyvyssään vastata hyvin erilaisten yleisöjen uutistarpeeseen ja avoimien rajapintojen kautta kumpaankin kysyntään vastaaminen helpottui. Avoimet rajapinnat osoittautuivat myös oivaksi työvälineeksi journalismin tutkimus- ja kehitystyölle. Toistaiseksi avoimet rajapinnat eivät olleet tutkituille uutistoimistoille suuri taloudellinen menestys, mutta siihen suurimmaksi syyksi tutkijat havaitsivat sen, että toiminta ei vielä ollut tarpeeksi avointa ja organisaatioilla ei ollut tarvittavia resursseja hyödyntää avointen rajapintojen tarjoamia etuja ja mahdollisuuksia. (Aitamurto & Lewis, 2013)

Otso Kivekäs kirjoittaa blogijulkaisussaan ”Toimittajaloukku ja kuinka se vältetään” rajapintojen olevan myös yksi ratkaisu Suomessa paljon pinnalla olleeseen toimittajaloukuongelmaan. Toimittajaloukulla tarkoitetaan tilannetta, jossa esimerkiksi kunta tai yritys tilaa ja ostaa tietojärjestelmän yritykseltä ja päätyy toimittajaloukkuun, jossa jokainen muutos ja kehitystarve on tilattava alkuperäiseltä toimittajalta, sillä muu ei ole mahdollista. Alkuperäinen järjestelmä on luotu niin, että muu kuin alkuperäinen toimittaja ei pääse sen toimintaan käsiksi ja näin tilaaja on sidottu asiakassuhteeseen niin pitkäksi aikaa kuin tietojärjestelmä on käytössä. Rajapintojen lisäksi ongelmaan tarjotaan ratkaisuksi esimerkiksi avointa lähdekoodia ja tietojärjestelmän pilkkomista eri toimittajille. (Kivekäs, 2012)

Avointen rajapintojen käytössä on myös mahdollisia ongelmia. Yrityksen maine saattaa kärsiä, mikäli avointa rajapintaa ei ole, tai mikäli se on toteutettu heikosti. Rajapinnan on oltava mahdollisimman virheetön ja toimintavarma saavuttaakseen käyttäjien suosion. Avoimen rajapinnan on oltava ehdottoman turvallinen käyttäjilleen ja toisaalta yrityksen on oltava varovainen, ettei avoimen rajapinnan kautta jaeta materiaalia, jota ei ole tarkoi-

tettu kolmansien osapuolien käyttöön. Rajapinnan julkaisijalla on koska tahansa valtuudet poistaa rajapinta kokonaan, rajoittaa sen avoimuutta tai päättää veloittaa rajapinnan käytöstä. Mikäli rajapinnan varaan on kehitetty omia toimintoja ja sovelluksia, voivat muutokset rajapintaan olla kova isku. Toisaalta rajapinnan käytön rajoittaminen saattaa vaikuttaa myös rajapintaa tarjonneen yrityksen maineeseen ja tämänkin takia, avoimen rajapinnan suunnittelu on tehtävä huolella. (Iazar, 2018)

Artikkeleiden ja aiemmin esitetyn aineiston perusteella voidaan todeta, että avoimet rajapinnat ovat tietoyhteiskunnassamme menestyksen kannalta hyvin olennainen tekijä ja niiden merkitystä ei sovi väheksyä. Avoimen rajapinnan julkaisu on useissa tapauksissa yritykselle kannattava ja suositeltava idea, mutta julkaisu on tehtävä tarkan harkinnan ja huolellisen valmistelun jälkeen.

2.1.3 Määritelmät

Avoim rajapinta voidaan määritellä eri yhteyksissä eri tavoilla. Suomen avoimien tietojärjestelmien keskus COSS ja Open knowledge Finland ovat olleet mukana kehittämässä virallista suomenkielistä määritelmää avoimille rajapinnoille. Tämän määritelmän on tarkoitus valmistuessaan palvella esimerkiksi kuntien tarpeita. (Kivekäs, 2014) Viimeisin versio määritelmästä on jo käytössä Suomen suurimpien kaupunkien yhteisessä 6Aika Avoin data ja rajapinnat -hankkeessa. (Varteva, 2016) Vuonna 2014 julkaistussa versiossa määritelmästä määritetään avoin rajapinta kolmella ehdolla, rajapinnan on oltava avoimesti dokumentoitu, käyttöön otettava ja testattava. Dokumentointi tarkoittaa, että rajapinnasta on avoimesti saatavilla kattava dokumentointi, joka on vapaasti käytettävissä. Käyttöön otettavuus tarkoittaa, että avoin rajapinta on mahdollista ottaa käyttöön ajankohdasta riippumatta ja ilman yhteydenottoja ylläpitäjään tai järjestelmätoimittajaan. Avoimen rajapinnan tulee olla myös testattavissa ajankohdasta riippumatta, kaiken datan ei kuitenkaan tarvitse olla avointa vaan testaamiseen riittää esimerkiksi rajattu testiaineisto. (Poikola et al., 2014)

Avoimen rajapinnan yksiselitteistä määritelmää tarvitaan esimerkiksi julkisissa järjestelmähankinnoissa. Mikäli avointa rajapintaa ei ole yksiselitteisesti määritelty, voidaan ajautua tilanteeseen, jossa järjestelmätoimittaja kertoo, että järjestelmässä on avoin rajapinta, mutta todellisuudessa dokumentaation saa vain ostamalla tai se on vain muutamien alihankkijoiden saatavilla. Rajapinta on avoin, jos sitä käyttävän sovelluksen kehittäminen on mahdollista kenelle tahansa. Tämä ei ole mahdollista pelkästään dokumentaation turvin, vaan rajapintaan on päästävä käsiksi ja sen avulla on pystyttävä testaa-

maan kehitettyjä sovelluksia. Sovelluksen kehittämisen on oltava mahdollista ilman erilisiä lupakyselyitä tai ilmoituksia siitä, mitä ollaan tekemässä tai kehittämässä. (Kivekäs, 2014)

Kattavien rajapintojen avulla voidaan saada tieto liikkumaan järjestelmien välillä entistä tehokkaammin ja tämä onkin useilla julkishallinnon organisaatioilla tavoitteena. Rajapintojen käyttö ei itsessään määrittele mikä on julkista tietoa ja mikä ei, mutta rajapintojen avulla tieto on mahdollista saada ulos järjestelmästä entistä ketterämmin. Rajapinta voi olla pelkkä datarajapinta tai toiminnallinen rajapinta. Datarajapinnan kautta saa nimensä mukaisesti siirrettyä dataa järjestelmästä toiseen ja järjestelmärajapinta mahdollistaa tiedon liikuttamisen lisäksi myös jonkin tai joitakin käsittelyjä tiedolle, esimerkiksi algoritmien kautta. Mikäli järjestelmällä on erilaisia rajapintoja, on täsmennettävä erityisen huolellisesti mikä kaikki on avointa. (Kivekäs, 2014)

Avoimen rajapinnan kautta saatava data on usein avointa, mutta se ei ole välttämätöntä. Esimerkiksi potilastietojärjestelmään tai muuhun terveydenhuollon järjestelmään voidaan rakentaa avoin rajapinta, vaikka potilaiden tiedot eivät ole julkisia. Avoin rajapinta ei automaattisesti tarkoita, että tuotantojärjestelmään tai sen tietoihin pääsisi kuka vain avoimesti käsiksi. Avoin rajapinnan pitää tarjota testiympäristö, mutta tieto voi olla esimerkiksi keksittyä esimerkkietoa. (Kivekäs, 2014)

Kuten jokaisessa ohjelmistoprojektissa, on rajapintojenkin suhteen oltava tarkkana sopimusehtojen kanssa. Esimerkiksi julkisen hallinnon IT-hankintojen yleiset sopimusehdot määrittelevät, että ostajan on halutessaan voitava avata rajapinta ilman toimittajan puutumista asiaan. Ehtoihin on tällöin kirjattava, että rajapinta on ostajan hallinnassa. Rajapintojen toteutuksessa on suositeltavaa käyttää yleisesti käytettyjä standardeja, protokollia sekä formaatteja. Avoin rajapinta ei velvoita järjestelmän omistajaa tarjoamaan pääsyä tuotantojärjestelmään tai tuotantodataan rekisteröitymättä käyttäjille, vaan heille voi olla tarjolla testijärjestelmä ilman todellista dataa. Avoin rajapinta suositellaan toteutettavaksi niin, että sen kuormittaminen ei aiheuta haittaa tietojärjestelmän muulle käytölle. Rajapintaan voidaan esimerkiksi asettaa tekninen raja kutsujen määrään aikayksikössä tai osa tietojärjestelmästä voidaan tarpeen vaatiessa kahdentaa. (Kivekäs, 2014)

2.2 Esimerkkejä käytännön sovellutuksista

Tässä kappaleessa esitellään erilaisia esimerkkejä avoimien rajapintojen käytännön sovelluksia eri toimialoilta. Avoimet rajapinnat ovat arkipäivää ja yhä useampi taho tarjoaa dataa julkisesti avoimen rajapinnan kautta. Esimerkiksi Suomen yleisradio, eli YLE on

vuodesta 2015 alkaen tarjonnut oman avoimen rajapintansa, jonka kautta on pääsy ohjelmätietoihin, videotallenteisiin ja näiden raportointiin (Yle api.). Suunnitteluohjelmistoistaan tunnettu Autodesk tarjoaa erilaisia vaihtoehtoja sovelluskehittäjille, joista osaan pääsee käsiksi vapaasti (ADN open - autodesk developer network.2019).

Toisaalta aina merkittävät avoimet rajapinnat eivät ole syntyneet yritysten tai yhteisöjen omasta vapaasta tahdosta, vaan esimerkiksi pankit ovat joutuneet kehittämään ja avaamaan teknisiä rajapintoja esimerkiksi tunnistautumispalveluita varten (Lehto, 2017). Myös esimerkiksi paikkatietojen läpinäkyvää arkistointia kuntien toimesta säädellään Euroopan Unionin alueella. Vaikka avoin rajapinta on vain yksi tapa toteuttaa direktiivin vaatimukset, on se suuri tekijä sille, miksi maanmittauslaitoksen ylläpitämä paikkatietoalusta on koko Suomen kattava paikkatietokanta. (Maanmittauslaitos,)

Yksittäisten ohjelmistojen ja yritysten tarjoamien avoimien rajapintojen lisäksi on olemassa suurempia avoimien rajapintojen kokonaisuuksia, joista muutamaa käsitellään seuraavaksi.

2.2.1 6Aika

6Aika on kuuden suurimman suomalaisen kaupungin yhteinen kestävä kaupunkikehityksen strategia, jossa mukana ovat siis Helsinki, Espoo, Vantaa, Tampere, Turku sekä Oulu. Strategian tavoitteena on vastata kaupungistumisen mukanaan tuomiin haasteisiin ja vastata kaupunkien asukkaiden todellisiin tarpeisiin älykkäämmillä palveluilla ja kaupungeilla. Strategian perustana ovat kolme suurta kärkihanketta: Avoimet innovaatioalustat, Avoin data ja rajapinnat ja Avoin osallisuus ja asiakkuus. Hankkeeseen on saatu rahoitusta mukana olevien kaupunkien lisäksi Euroopan aluekehitysrahastolta ja Suomen valtiolta. (Älykaupungit tehdään yhdessä.2014)

Strategian puitteissa on julkaistu myös opas ”Avoimet ja yhtenevät rajapinnat – Avain digitaalisiin palveluihin”, jossa listataan avoimien rajapintojen hyötyjä kaupungeille, asukkaille ja yrityksille. Oppaassa kuvaillaan kaupunkien käytössä olevat erilaiset rajapintatyytit ja annetaan selkeät ohjeet, kuinka luoda strategian mukainen toimiva rajapinta. Kaikki mukana olevat rajapinnat on pyritty harmonisoimaan, jotta avoimien rajapintojen käyttö on mahdollisimman mutkatonta ja välitettävä tieto pysyy yhdenmukaisena. (Var-teva, 2016)

Kaupungit ajatellaan perinteisesti rakennusten ja katujen muodostamiksi fyysisen ympäristön kokonaisuudeksi, mutta jatkuvasti enenevässä määrin kaupunkeja luodaan, johdetaan ja kehitetään digitaalisilla välineillä. Avoimilla rajapinnoilla tuotetaan kaupunkien

asukkaille digitaalisia palveluja, joiden neljä suurinta lohkoa ovat tässä strategiassa: palauterajapinnat, tapahtumarajapinnat, päätösrajapinnat ja resurssienvarausrajapinnat. (Varteva, 2016)

Palauterajapinta antaa jokaiselle kaupunkilaiselle vaivattoman tavan antaa palautetta yhteisen ympäristön puutteista. Rajapinnan avulla on mahdollista myös seurata palautteen aiheuttamia toimenpiteitä. Kaupunkilaisten ja virkamiesten yhteydenpito muuttuu rajapinnan avulla suoraksi ja avoimeksi. Tapahtumarajapinta auttaa kaupunkilaisia löytämään kiinnostavat tapahtumat yhdestä paikasta. Tapahtumien tiedot on mahdollista hakea yhdestä rajapinnasta, joka edesauttaa tietojen paikkansapitävyyttä eri julkiasu- alustoissa, rajapintaa saa vapaasti käyttää myös kaupallisiin tarkoituksiin (Varteva, 2017). Päätösrajapinnalla haetaan lisää läpinäkyvyyttä kaupunkien toimintaan julkaisemalla esimerkiksi kaupunginhallituksen ja eri lautakuntien esityslistat ja pöytäkirjat yhdessä paikassa, josta jokaisen on mahdollista niitä tarkastella. Resurssienvarausrajapinta tuo näppäryyttä ja kannattavuutta lyhytaikaiseen vuokraukseen. Yhdestä paikkaa löytyy esimerkiksi kaikki kaupungin ja yksityisten toimijoiden erilaiset vuokrattavat urheilutilat. Tarkoitus ei ole rajata rajapintaa vain kaupungin omistamien tilojen varauksiin, vaan elävöittää koko kaupunkia. (Varteva, 2016)

2.2.2 Pankkien avoimet rajapinnat

Kaikissa tapauksissa innoitus rajapintojen avaamiseen ei lähde yritysten sisältä, asiakailta tai palveluiden käyttäjiltä, vaan rajapintoja saatetaan vaatia myös ylemmiltä tahoilta. Vuoden 2018 alusta EU:ssa (Euroopan Unionissa) voimaan tullut maksuliike- ja palveludirektiivi vaatii pankkeja avaamaan rajapintoja kolmansille osapuolille. Rajapintojen kautta pankkien on tarjottava pääsy tunnistautumispalveluihin sekä tilitietoihin ja tilitapahtumiin asiakkaiden niin halutessa. (Lehto, 2017)

Pankkien tarjoamat tunnistautumispalvelut rajapintojen kautta kolmansien osapuolien käyttöön tarjoavat turvallisemman ja luotettavamman tavan tunnistautua ja liittää pankkitietonsa erilaisiin palveluihin. Vanha perinteinen tapa, jossa asiakas tallentaa pankkitietonsa kolmannen osapuolen palveluun sisältää enemmän riskejä väärinkäytöstä, jotka eivät direktiivin myötä täysin poistu. Rajapintojen avulla on kehitetty myös pankeille toimintaa helpottavia ja tehostavia ohjelmistoja, joilla on esimerkiksi pyritty helpottamaan rahanpesuun liittyvän toiminnan havaitsemista. (Jackson, 2018)

Rajapintojen avaaminen luo uusia mahdollisuuksia, mutta direktiiville on myös selkeä negatiivinen puoli. EU tai muutkaan tahot eivät korvaa pankeille direktiivistä aiheutuvia kuluja, vaan kulut päätyvät asiakkaiden maksettavaksi muiden pankkikulujen joukossa.

Yhtenä perusteluna direktiiville on käytetty kansainvälisen pankkikilpailun edistämistä. Direktiiviä kritisoidaan siitä, että se asettaa yrityksiä eri asemiin huomioimatta sitä, että kaikki pankit eivät välttämättä koe tarvetta osallistua kansainväliseen kilpailuun asiakkaista, mutta joutuvat silti investoimaan rajapintojen avaamiseen kuten direktiivistä hyötyvät pankit. (Lehto, 2017)

2.2.3 Paikkatiedot

Suurelle yleisölle yksi näkyvimpiä avoimen rajapinnan sovelluksia on Google Mapsin päälle rakennettu kartta lähes minkä tahansa yrityksen kotisivuilla. Eräiden lukujen mukaan Googlen karttapalveluita käyttää joka viikko yli kaksi miljoonaa erillistä sovellusta. (Varteva, 2017)

Karttoja ja paikkatietoja on saatavilla runsaasti erilaisiin käyttötarkoituksiin erilaisten rajapintojen kautta. Maanmittauslaitos tarjoaa karttoja niin retkeilijöille, kuin paikkatietoa ammattikäyttöön. Samalle sivustolle on kerätty paikkatietoja yli organisaationrajojen. Tiedostopalvelun kautta on mahdollista tilata itselleen valitsemiltaan alueilta niin laserkeilausaineistot, korkeusmallit kuin kiinteistörekisterikartatkin, joita hyödynnetään esimerkiksi maastomallien luonnissa. (Kartat ja paikkatieto.)

Maanmittauslaitoksen ylläpitämä paikkatietoalusta palvelee kuntia myös INSPIRE-velvoitteiden täyttämässä. INSPIRE-direktiivi on EU-direktiivi, joka edesauttaa yhteiskäyttöisiä paikkatietoja ja paikkatietoinfrastuktuuria. Tuomalla paikkatietoalustaan osoitteet, rakennukset, ajantasa-asemakaavat ja yleiskaavat, kunnat täyttävät velvoitteensa. Direktiivillä pyritään vähentämään päällekkäistä työtä tietojen ylläpidossa ja alentamaan kustannuksia ja toisaalta takaamaan tiedon yhtenäisyys. (Maanmittauslaitos,)

Väylävirasto tarjoaa kehittäjille kaksi erilaista ylläpitämäänsä paikkatietorajapintaa, WMS (Web Map Service) -rajapinnan, jonka kautta on mahdollista ladata valmiina karttakuvina paikkatietoaineistoja ja WFS (Web Feature Service) -rajapinnan, joka tarjoaa paikkatietokohteiden tietoja vektorimuodossa. Lisäksi kehittäjille on tarjolla toiminnallisia rajapintoja, joilla voi esimerkiksi muuntaa koordinaatteja tie- ja katuosoitteiksi tai tieosoitteita toisen ajanhetken tieosoitteiksi. (Kehittäjille.)

3. TEKLA STRUCTURES OHJELMISTO

Tekla Structures on Trimble Solutions Corporationin rakennusten tietomallintamiseen keskittyvä ohjelmisto erityisesti rakennesuunnittelijoiden käyttöön. Tekla Structures noudattaa BIM (Building Information Mode) teknologiaa, joka tehostaa ja helpottaa rakennesuunnittelua liittämällä laskennan, rakennemallit, dokumentoinnin ja piirustukset yhdeksi kokonaisuudeksi tarjoten mahdollisuuden jakaa tietomalli kaikkien rakennusprojektin osapuolten kesken. (Trimble Solutions Corporation,)

3.1 Tekla Open API

Tekla Open API on Trimblen Tekla ohjelmistolle tarjoama avoin rajapinta. Tekla Structu-resin käyttöliittymä tarjoaa ihmiskäyttäjille mahdollisuuden kommunikoida ohjelmiston kanssa. Teklan Open API, joka tunnetaan myös termillä .NET API, eli Teklan avoin rajapinta, tarjoaa vastaavasti muille ohjelmistoille mahdollisuuden kommunikoida ohjelmiston kanssa. Avoin rajapinta mahdollistaa kolmannen osapuolen sovelluksille kommunikoinnin mallien piirustusobjektien kanssa ja erilaisten laajennusosien luomisen. Tekla Open API toimii Microsoft .NETin 4.0 Frameworkin täydessä versiossa. (Tekla open API developer's guide2017)

Tässä kappaleessa esitellään Tekla Structures ohjelmiston avointa rajapintaa ja kuinka rajapinta toimii. Ensimmäisessä luvussa avataan ohjelmistotuotanto-termiä ja ohjelmiston elinkaarta. Sen jälkeen kerrotaan, miksi rajapintaa tarvitaan, rajapinnan käyttöympäristöstä ja tarvittavasta ohjelmointikielestä.

3.1.1 Ohjelmistotuotanto

Ohjelmistotuotanto tai ohjelmistotekniikka ovat molemmat käännöksiä englannin kielelle termille Software Engineering. Vuonna 2004 julkaistussa kirjassaan nimeltä Ohjelmistotuotanto Haikala ja Märijärvi määrittelevät ohjelmistotuotannon tarkoittavan ohjelmistotyötä. Ohjelmistotyön tuloksena syntyy järjestelmä, jonka voidaan olettaa toteuttavan käyttäjiensä toiveet ja odotukset määrittelyjen mukaisesti. Järjestelmän on myös määritelmän mukaisesti valmistuttava ennalta laadittujen aikataulujen ja kustannusarvioiden puitteissa. Ohjelmistotuotannon voidaan siis ajatella luovan kuria ohjelmiston kehitysprosessiin. (Haikala & Märijärvi, 2004)

Rakennusalalla on viime vuodet ollut nousussa trendi, jossa erityisesti toimistorakennuksista tehdään muuntojoustavia. Muuntojoustavuudella tarkoitetaan, että rakennuksen ti-

loja on helppo muokata tulevaisuudessa vastaamaan uusia käyttötarpeita. Rakennusala on vanha ala verrattuna, ohjelmistoalaan, mutta muuntojoustavuus on ohjelmistopuolella jo vakiintunut tapa ja sen historia on pidempi kuin rakennusosalalla. Ohjelmistotuotannossa on hyvin tavallista muunnella valmista tuotetta. Tilastojen mukaan jopa vain noin kolmasosa alalla tehtävästä työstä on uusien ohjelmistojen kehittämistä ja kaksi kolmasosaa on vanhojen ohjelmien ylläpitotyötä. Ylläpitotyöllä tarkoitetaan vanhojen ohjelmien muuttamista, korjaamista ja jatkokehitystä. (Haikala & Märijärvi, 2004)

Lähtökohtana kehittymiselle on käytännön ongelma, jolle etsitään ratkaisua. Aluksi ongelman ratkaisuksi kelpaa mikä tahansa toimiva tapa. Kun tietotaito kehittyy systemaattisemmaksi, se kiteytetään heuristiikoiksi ja työskentelymenetelmiksi. Heuristiikalla tarkoitetaan menetelmää, jolla päästään tehokkaasti tarpeeksi lähelle parasta mahdollista lopputulosta. Ohjelmistokehityksessä käytetään usein myös niin sanottua ad hoc -menetelmää, jossa kulloinkin kyseessä olevaan ongelmaan etsitään ratkaisu. Tekniikoiden kehittyessä löydetään yhä uusia ja monimutkaisempia ongelmia ja kehitys jatkuu. (Haikala & Märijärvi, 2004)

Helsingin Yliopiston Tietojenkäsittelytieteen laitoksen luentomonisteen mukaan ohjelmistoprosessi, eli ohjelmistotuotanto, harvoin päättyy, kun ohjelmisto, eli tuote toimitetaan asiakkaan käyttöön. Ohjelma tarvitsee myös ylläpitoa, jota usein korjataan sitä mukaa kun virheitä löytyy, ohjelmaan lisätään uusia toiminnallisuuksia käyttäjän tarpeiden mukaan ja joitain osia ohjelmasta saatetaan jopa kirjoittaa kokonaan uudestaan. (Luukkainen & Laine, 2012) Avoimen rajapinnan kautta käyttäjä voi itse osallistua ohjelmistotuotannon ylläpitovaiheeseen itse lisäämällä toiminnallisuuksia ja muokkaamalla jo olemassa olevia toiminnallisuuksia.

3.1.2 Käyttöympäristö

Teklan Open API on kehitetty ohjelmistokomponenttikirjasto Microsoft .NET:iä hyödynnäen. Kaikki Trimblen tarjoamat koodiesimerkit on toteutettu käyttäen Microsoftin Visual Studio ohjelmankehitysympäristöä käyttäen C# ohjelmointikieltä. (Tekla open API developer's guide2017)

C# on olio-ohjelmointiin perustuva ohjelmointikieli, joka on luotu C++ ja Javan yhdistelynä, tarkoituksena luoda helppokäyttöinen ohjelmointikieli. C# on erityisen toimiva verkossa tapahtuvaan olio-ohjelmointiin, jolloin kokonainen tietoverkko voidaan nähdä tietokoneena ja verkon eri osissa toimivat ohjelmat keskustelevat keskenään. Olio-ohjel-

moinnissa olio yhdistää datan ja siihen liittyvät operaatiot. Ohjelman toiminnallisuus perustuu useiden olioiden kommunikointiin keskenään tehtäväpyyntöjen, eli viestien välityksellä. (Paananen & Granlund, 2005)

C# on alusta alkaen rakennettu olio-ohjelmointiin ja helpottamaan komponenttien luomista, joita voidaan helposti jälleen käyttää ohjelmoinnissa. Periyttäminen on avain ominaisuus olio-ohjelmoinnin paradigmassa. Voimme poistaa luokkien yhteiset piirteet ja sijoittaa ne korkean tason kantaluokkaan. Voimme lisätä tai muuttaa ominaisuuksia erityisissä periytetyissä luokissa, jotka perivät standardikäyttäytymisen kantaluokastaan. Perinnöllisyys edesauttaa koodien uudelleen käyttöä ja laajennettavuutta. (Oberg, 2002)

Sovelluksen kirjoittaminen voi tapahtua yksinkertaisesti samaan tapaan kuin millä tahansa tekstieditorilla ja kääntämällä erillisellä kääntäjäohjelmalla, jonka jälkeen konekielinen ohjelma voidaan suorittaa. Kehittyneempi tapa on käyttää integroitua ohjelmointiympäristöä IDE:ä (Integral Development Environment), joka on ohjelmisto, joka sisältää editorin, kääntäjän ja virhejäljittimen. Kehittynein tapa on käyttää sovelluskehitysympäristöä, eli sovelluskehittäjää. Näissä ohjelmat koostetaan hyvin pitkälti leikkaa ja liimaa-tekniikalla valmiista komponenteista graafisessa ohjelmassa. Ohjelmoija voi räätälöidä kehittämissä olevia valmiita komponentteja ja näiden toimintaa, ja määritellä itse uusia komponentteja ja muuta toiminnallisuutta. Yksi tällainen kehitysympäristö on Visual Studio, jota myös Tekla Open API hyödyntää. (Paananen & Granlund, 2005)

Ohjelmistoarkkitehtuuri on kokonaisuus standardien mukaisia tai muuten yleisesti hyväksytyjä (eli de facto -standardeja) ohjelmistoteknologioita, jotka on kehitetty toimimaan yhdessä. Näin pyritään varmistamaan erilaisten järjestelmien, laitteiden ja palveluiden yhteensopivuus. Ohjelmistoteknologiaan kuuluvat yhteensopivat palvelinohjelmit, verkko-ohjelmit, ohjelmointikielet ja kommunikointimenetelmät. (Paananen & Granlund, 2005)

Microsoft .NET on Microsoftin visio sovellusten tulevaisuudesta Internetin aikakaudella. Se tarjoaa parannettuja yhteentoimivuusominaisuuksia perustuen avoimiin internets-tandardeihin. Ohjelmistokehittäjille .NET tarjoaa ohjelmointialustan hyvillä ja helppokäyttöisillä työkaluilla. Microsoftin .NET on harppaus ohjelmointitekniikassa, joka helpottaa erityisesti applikaatioiden kehittämistä. (Oberg, 2002)

.NET framework, eli työskentely- tai kehitysympäristö, on ympäristö, jonka käytön aloittamiseen ei tarvitse tietää paljon ohjelmoinnista. Ohjelma kirjoitetaan kehittyneellä, selkokielisellä ohjelmointikielellä ja compiler, eli kääntäjä, tekee siitä .EXE-tiedoston (Executable), jonka ohjelmoija voi ajaa. .Net Framework luokkakirjasto tulee .NET ympäristön mukana ja sitä tarvitaan yksinkertaisimpienkin ohjelmien luonnissa. Ympäristön luo-

ma .exe tiedosto ei sisällä suoritettavaa koodia, vaan koodi on IL (Intermediate Language) välikielellä ja windows ympäristössä tämä koodi pakataan suoritettavaan formaattiin, jolloin käyttäjälle näkyy tuttu exe- tiedosto tai komponentteja luotaessa dll-tiedosto. Kun koodi suoritetaan, erityinen CLR (Common Language Runtime) -ympäristö käynnistetään ja IL käskyt suoritetaan CLR:n toimesta. Erityistä on, että IL:ää ei tulkita joka kerta kun se suoritetaan, vaan CLR tulee JIT (just-in-time) kääntäjän kanssa ja kääntää IL:n natiiviksi konekieleksi (native machine code). (Oberg, 2002)

.NET ympäristö koostuu CLR:stä, .NET Framework Class Librarystä, CLS:stä (Common Language Specification), monista .NET ohjelmointikielistä ja Visual Studio.NET:stä. CLR (runtime tarkoittaa aikaa, kun ohjelma ajetaan) tarjoaa palveluita ohjelmien suorittamiseen. NET Framework Class Library sisältää yli 2500 classia (luokkaa) joiden toiminnallisuudet ovat saatavilla kaikille .NET ohjelmointikielille. CLS määrittelee säännöt ominaisuuksille ja käyttökäytännöille. Tyypit ovat CLR:n ohjelmistomallin sydän. Tyyppi on käytännössä luokka suurimmassa osassa olio-ohjelmointikieliä. (Oberg, 2002)

Assembly eli kokoonpano on ryhmittely tyypeille ja resursseille, jotka toimivat yhdessä loogisena yksikkönä. Kokoonpano voi olla looginen DLL tai EXE. Kokoonpano koostuu yhdestä tai useammasta tiedostosta, eli moduulista, jotka voivat olla EXE tai DLL tiedostoja tai resursseja. Kokoonpano sisältää sen lisäksi metadatan, jossa mainitaan tärkeimmät keskinäiset riippuvaisuudet ja versiotiedot. (Oberg, 2002)

EXE on lyhenne toteutettavasta tiedostosta. EXE-tiedostoja käytetään käyttöjärjestelmien sisällä ohjelmien avaamiseen. EXE-tiedostot voivat olla itsestään purkautuvia. EXE-tiedostot viittaavat usein niihin liittyviin DLL-tiedostoihin. (Fisher, 2018)

DLL on lyhenne sanoista dynamic link library, eli vapaasti suomennettuna dynaaminen linkkikirjasto. DLL tiedostojen käyttö edesauttaa koodien jatkokäyttöä, kierrätettävyyttä ja tehokasta muistinkäyttöä. DLL takia ohjelmat latautuvat ja toimivat nopeammin, ja vaativat vähemmän muistitilaa. Ongelmana käytössä voivat olla riippuvuudet, joiden takia ohjelma ei välttämättä toimi. Kun ohjelma käyttää DLL tiedostoa, syntyy riippuvuus ja mikäli siihen aiheutuu häiriötä, ohjelman toiminta saattaa pysähtyä kokonaan. Ongelmia voi myös ilmetä, jos toinen ohjelma yliajaa tai pysäyttää toiminnan ohjelmien välisen ristiriitojen takia. Assemblyjä, eli kokoonpanoja käyttämällä suurin osa riippuvuusongelmista voidaan ratkaista, koska silloin riippuvuudet talletetaan paikallisesti ja ohjelma ei ole enää riippuvainen ulkopuolisista lähteistä. (What is a DLL.2018)

DLL tiedosto on kirjasto, joka sisältää koodia ja dataa, joita useampi kuin yksi ohjelma voi käyttää samanaikaisesti. DLL:ien avulla tiedostot voidaan modularisoida, eli pilkkoa

koodi yksittäin käytettäviin osiin, erillisiksi komponenteiksi, jolloin ohjelmaa voidaan päivittää komponentti kerrallaan esimerkiksi hintojen tai standardien päivittyessä. DLL etuja on, että se vaatii vähemmän resursseja (What is a DLL.2018)

Suuret modernit sovellukset ovat harvoin yhtenäisiä ja koostuvat vain yhdestä EXE tiedostosta. .NET:ssä ne koostuvat useista suoritettavista yksiköistä. .NET:in Ympäristössä sovellus koostuu yleensä yhdestä EXE tiedostosta ja lukuisista DLL tiedostoista. (Oberg, 2002)

3.2 Open API-toiminnot

Trimble tarjoaa Tekla Structures kehittäjille käyttöön avoimen rajapinnan, jonka avulla luoda lisäosia valmiiseen ohjelmaan. Tekla Open API:lla kehitettyjä Tekla Structuresin omia sovelluksia ja lisäosia kutsutaan laajennuksiksi. Open API:n hyödyntäminen vaatii ohjelmointia Tekla Structuresin ulkopuolella, mutta Tekla Open API:n mahdollisuuksista on mahdollista hyötyä myös kokonaan ilman ohjelmointitaitoja hyödyntämällä Tekla Warehousea, josta voi ladata muiden kehittämiä laajennuksia. (Develop applications using tekla open API.2019)

Tekla Open API:n avulla voidaan esimerkiksi tallentaa ja suorittaa käyttöliittymän erilaisia toimintoja, joiden avulla voi automatisoida erilaisia tehtäviä kuten päivittäistä raportointia. Voidaan luoda automatisoituja työkaluja toistuviin tyypillisiin kohteisiin, kuten tavallisimpiin rakennesein tai toistuviin yksityiskohtiin piirustuksissa. Integroida Tekla Structures osaksi muita ohjelmistoja, esimerkiksi analyysityökaluja tai hyödyntää automaattista tiedonsiirtoa ohjelmistojen välillä. Tekla Open API:n avulla on mahdollista myös kehittää täysin uusia omia tarpeita vastaavia toimintoja ja ominaisuuksia. (Develop applications using tekla open API.2019)

Tekla Structuresin avoimen rajapinnan hyödyntämismahdollisuudet voidaan jakaa kolmeen tyyppiin, applikaatioihin, eli sovellutuksiin, plugineihin ja custom propertyihin. Avoimen rajapinnan avulla työkalujen luomista helpottamaan Trimble tarjoaa laajan referenssikirjaston. (Tekla open API developer's guide2017)

Teklan avoimen rajapinnan voidaan ajatella myös jakautuvan neljään erilliseen rajapintaan: macro, malli, plugin ja piirustus rajapintoihin. Macro rajapinnan avulla tallennetaan ja toistetaan kommentoja ja valintaikkunoiden toimia. Tallennettuja macroja voidaan myös jälkikäteen muokata koodin avulla. Malli rajapinnassa luodaan ja muokataan mallin objekteja ja niiden tietoja. Rajapinta on yhteydessä kulloinkin käynnissä olevaan malliin, josta voidaan esimerkiksi luoda halutunlaisia näkymiä. Piirustus rajapinnassa luodaan ja muokataan piirustuksia ja piirustusten objekteja. Rajapinnan kautta pääsee myös käsiksi

esimerkiksi piirustusluetteloon. Plugin rajapinnan avulla luodaan älykkäitä objekteja mallissa, tällaisia voivat olla esimerkiksi portaat ja käsijohteet. (Trimble Solutions Corporation,)

Tekla Open API:n avulla on mahdollista myös siirtää tietoa suoraan muiden ohjelmistojen rajapintoihin. Tämä tapahtuu malli rajapinnan avulla. Tärkeää siirtää kaikki olennainen tieto kuitenkin kuormittamatta ylimääräisellä tiedolla. (Trimble Solutions Corporation,)

Alla olevaan taulukkoon 1. on kirjattu kolmen eri laajennustyyppin, Custom Property, Pluginin ja Applikaation erilaiset ominaisuudet.

Taulukko 1. Tekla Open API-toiminnot ja niiden ominaisuudet (Tekla open API developer's guide2017)

	Custom Property	Plugin	Applikaatio
Erillinen objekti	Kyllä	Kyllä	Ei
Automaattinen päivitys	Kyllä	Kyllä	Ei
Integroitu valintaikkuna	Kyllä	Kyllä	Ei
Suorituskyky	Hidas	Nopea	Kohtuullinen
Muisti	Runsas	Vähäinen	-
Joustavuus	Matala	Runsas	Kohtuullinen
Vaikeustaso	Matala	Kohtuullinen	Kohtuullinen

Seuraavissa kappaleissa tutustutaan tarkemmin eri laajennustyyppeihin, joita avoimen rajapinnan kautta voidaan toteuttaa.

3.2.1 Applikaatiot

Applikaatiot eli sovellukset ovat yksinkertaisin tapa hyödyntää Tekla Open API:a ja tästä syystä kaikki Trimblen tarjoamat harjoitustehtävät ovat applikaatioita. Harjoitustehtävissä ohjelmoidaan esimerkiksi graafisia käyttöliittymiä, joilla voidaan luoda avoinna olevaan Tekla Structures malliin esimerkiksi anturat ja näille pilarit sekä raudoitukset. Trimblen tarjoamien laajojen referenssikirjastojen ja selkeiden ohjeiden avulla yksinkertaisten sovellusten ohjelmoiminen on pyritty tekemään mahdollisimman helpoksi käyttäjille. Sovellus suoritetaan erillisessä prosessissa eikä käynnistetä suoraan Tekla Structuresin sisällä, ellei sovellusta ole upotettu makroon. (Trimble Solutions Corporation, 2018)

3.2.2 Pluginit

Pluginit eli niin sanotut liitännäiset tai laajennukset ovat komponenttityökaluja, joka tarkoittaa, että ne ovat muokattavia ja riippuvia syötetyistä lähdeobjekteista. Pluginit voidaan ladata ja käynnistää Teklan sisällä. Plugin on mukautettu kokonaisuus mallissa,

jolla on oma valintaikkuna. Malli pluginit valitaan komponenttikatalogista ja piirustusten pluginit saadaan käyttöön työkaluriviltä ja niiden avulla luodaan malliin tai piirustukseen uusia objekteja. Plugineita voivat olla esimerkiksi erilaiset liitokset tai muut yksityiskohdat. (Trimble Solutions Corporation,)

Kun käyttäjä avaa pluginin käynnistyy konstruktorimenetelmä, joka pyytää käyttäjältä tarvittavat lähtötiedot dialogi-ikkunan kautta. Plugin soveltaa arvoja ja tulostaa halutun objektin. Käyttäjän antamat lähtöarvot ja rakenteen tiedot tallennetaan Teklan mallitietokantaan. Käyttäjille pluginit ovat graafisten käyttöliittymiensä ja valintaikkunoidensa ansiosta selkeitä käyttää. (Trimble Solutions Corporation,)

Pluginit eivät voi muokata lähtötietojaan itsenäisesti, mutta niille voidaan määrittää attribuutti, eli ominaisuus, joka tunnetaan termillä `InputObjectDependency` eli vapaasti suomennettuna tarkoittaa lähtötiedon, eli syötteen, ja objektin riippuvuussuhdetta. Attribuutin määrittelyksi voidaan antaa jokin neljästä vaihtoehdosta:

- Riippuvainen, eli plugin päivittyy, kun syöte muuttuu.
- Riippumaton, jolloin plugin ei päivity, vaikka syöte muuttuu.
- Geometrisesti riippuvainen, eli plugin päivittyy kun syötteen geometriaa muutetaan.
- Muokattava riippumaton, eli plugin on riippumaton syötteestä, mutta sitä on mahdollista muokata mallissa, jolloin luodut objektit ovat suhteessa pluginiin. Plugin valintaikkuna voidaan avata valitsemalla objekti mallissa ja ikkunan kautta objektin arvoja voidaan muokata.

Pluginien avulla luotuja objekteja on helppo muokata valintaikkunan kautta, järjestelmä muokkaa automaattisesti olemassa olevia osia eikä luo esimerkiksi uusia eri kokoisia osia malliin. Pluginien voidaan ajatella elävän mallin sisällä ja niillä voidaan usein korvata Custom Componentteja, joskin pluginit vaativat kehittäjiltään ohjelmointitaitoja, esimerkiksi C#-kielellä. (Trimble Solutions Corporation,)

Pluginien tarkoitus on tuoda ohjelmistoon yksittäinen lisäominaisuus tai saada ohjelmisto toimimaan jollakin tapaa paremmin. Tekla Open API:ssa pluginit ovat tehokkain keino muokata ohjelmistoa omiin tarpeisiin, mutta samalla myös vaativin sekä ohjelmointitarpeiden, että Tekla mallin toimintaperiaatteiden ymmärtämisen osalta. Yksinkertaisimpiin tarpeisiin pluginit ovat usein turhan monimutkainen ratkaisu. (Trimble Solutions Corporation,)

3.2.3 Custom property

Custom Propertyt, eli vapaasti suomennettuna muokatut ominaisuudet, ovat ohjelmoituja työkaluja, joilla lasketaan malliarvoja, joita käytetään raportoinnissa ja piirustuksissa. Custom Propertyt mahdollistavat ulkoisen laskennan. Ne ladataan ohjelman sisällä ja niitä kutsutaan syntaksilla CUSTOM.xxx samalla tavalla kuin muitakin malliasetuksia. (Trimble Solutions Corporation, 2018)

3.3 Uusimmat työkalut ja kehitteillä olevat mahdollisuudet

Trimble tarjoaa tukea ja vinkkejä käyttäjilleen nettisivujensa kautta. Iso osa materiaalista, kuten keskustelufoorumi, on saatavilla vain rekisteröityneille käyttäjille, rekisteröityminen on ilmaista ja vaatii vain toimivan sähköpostiosoitteen. Keskustelufoorumilla on tarjolla tukea paitsi muilta käyttäjiltä, myös Trimblen asiantuntijoilta. Osa materiaaleista on saatavilla vain asiakasyrityksille.

Seuraavaksi tutustutaan tarkemmin Trimblen viime vuonna Suomen markkinoille tuomaan Tekla Tedds mitoitusohjelmistoon ja Trimblen tarjoamaan Tekla Developer Centeriin sovelluskehittäjille.

3.3.1 Tekla Tedds

Tekla Tedds on viime vuonna Suomen markkinoille julkaistu perinteinen englantilainen laskentaohjelmisto. 12.10.2018 Trimblen Account Manager Minna Arola kertoo ohjelmiston mahdollisuuksista webinaarissa ”Tekla Tedds – luotettava ohjelmisto laskelmien tekemiseen. Webinaari on katsottavissa ilmaiseksi rekisteröityneille käyttäjille. (Arola, 2018)

Tekla Tedds on vasta 2018 julkaistu Suomen markkinoille uudistuksen jälkeen, jossa lisättiin laskentanormiston osaksi eurokoodi ja kansalliset liitteet. Ohjelmisto on tarkoitettu toistuvien rakenneosien mitoitukseen, korvaamaan perinteisiä excel-laskentapohjia ja käsin laskentaa. Ohjelmisto korvaa myös analyysityökalut 2D-kehien tarkasteluissa, sillä sen taustalla pyörii FEM-laskentamoottori (Finite Element Method). Ohjelmistosta löytyy valmiita laskentapohjia teräkselle, betonille ja puulle, valmiit laskentapohjat on ryhmitelty sekä materiaaleittain, että rakenneosittain. Ohjelmistoa markkinoidaan sillä, että yhdellä lisenssimaksulla käyttäjä saa käyttöönsä lähes kaikki käytössä olevat laskentanormit, sillä ohjelmiston laskentakirjastosta löytyvät kaikki suurimmat normit aina eurokoodista Kanadan ja Yhdysvaltojen vastaaviin. Ohjelmisto päivittyy tiuhaan ja takaa käyttäjille aina ajantasaiset laskentanormit. (Arola, 2018)

Ohjelmisto on helppokäyttöinen ja käyttövalmis ilman pitkää ohjelmiston opiskelua. Suurimmaksi eduksi mainitaan ohjelmistosta suoraan tulostettavat raportit, joista käyttäjä saa valita tulostetaanko raportti täyspitkänä vai tiivistettynä ja mitä tietoja raporttiin tuodaan. Raportit ovat keskenään helposti vertailtavissa ja helpottavat ulkopuolista tarkastusta ja toisaalta myös tarkastusta yrityksen sisällä. Laskenta säilyy läpinäkyvänä, kun raportista on helppo tarkistaa mistä kohtaa eurokoodia mikäkin laskentakaava on. Ohjelmistossa työskennellään projektikohtaisesti, jolloin kaikki projektin laskelmat löytyvät yhdestä paikkaa ja näin laskelmiin on helppo palata tarvittaessa myöhemminkin. Laskelmia on myös helppo monistaa projektin sisällä, mikäli tarvitaan esimerkiksi eri kuormituksilla useita samanlaisia rakenneosia tai halutaan tuoda vanhan projektin onnistuneet laskelmat osaksi uutta projektia. Ohjelmistossa on mahdollista tehokkaasti tutkia erilaiset kuormitustapaukset ja -yhdistelyt. (Arola, 2018)

3.3.2 Tekla Developer Center

Tekla Developer Center on Trimblen tarjoama Tekla-kehittäjäkeskus, jonne on koottu kaikki mahdollinen tieto Teklan Open API:sta. Sivustolta löytyvät niin avoimen rajapinnan dokumentaatio kuin pääsy Trimblen tarjoamaan harjoittelumateriaaliin. Harjoitusmateriaalin avulla kuka tahansa voi tutustua Teklan avoimen rajapinnan tarjoamiin mahdollisuuksiin. Sivustolle on koottu myös paljon esimerkki koodeja ja muuta materiaalia. (Tekla developer center.2019)

Sivustolta löytyvät myös ohjelmointioppaat ja kehittäjille on tarjottu mahdollisuus olla mukana parantamassa rajapinnan toimintaa raportoimalla sivuston kautta suoraan löytämistään virheistä. Lisäksi kehittäjien on mahdollista osallistua Teklan avoimen lähdekoodin kehitystyöhön. Asiakasyritystensä Tekla-kehittäjille tarjolla on keskustelupalstojia, joiden kautta on saatavilla tukea kehitystyöhön niin Trimblen osajilta kuin myös muilta Tekla-kehittäjiltä. Sivuston kautta on myös mahdollista jakaa omia kehitystöitään muiden käyttäjien testattavaksi ja jatkokehittettäväksi. Uutena ominaisuutena sivustolle on lisätty hakemisto, johon Teklan kehitystyöhön erikoistuneet yritykset ja henkilöt voivat jättää yhteystietonsa, ja kehitystyöhön tekijöitä etsivät voivat sivuston välityksellä etsiä tarpeisiinsa sopivaa asiantuntijaa. (Tekla developer center.2019)

4. SELVITYS OPEN API-SOVELLUSTEN KÄYTÖSTÄ KOHDEYRITYKSESSÄ

Tässä luvussa kerrotaan Tekla Structuresin avoimen rajapinnan hyödyntämistä kohdeyrityksessä nykytilassa ja tarpeista tulevaisuudessa. Selvitystyö aloitettiin tutkimalla intraa, verkkolevyjä ja henkilökunnan keskustelualustaa yammeria. Tietokonetta tutkimalla löytyi vain Tekla Structuresia koskevia tiedostoja sekä muutama tiedosto ja ohje Autodeskin Robot Structural Analysis -ohjelmistoa varten. Tutkimusmenetelmän heikkoutena on, että etsinnöissä saattoi jäädä jotain huomaamatta ja osaan keskustelualueista sekä verkkolevyistä on rajattu käyttäjäjoukko. On todennäköistä, että muiden yksiköiden ja liiketoiminta-alojen osalta tieto ei ole täysin kattava, koska tiedot näiden käyttämistä ohjelmistoista perustuivat oletuksiin.

Selvitystyötä jatkettiin keskustelemalla kohdeyrityksen BIM manager Teemu Ahosen kanssa ja toteuttamalla kohdeyrityksen Tampereen rakennesuunnitteluyksikössä kyselytutkimus. Näiden perusteella määritetään tavoitteet avoimen rajapinnan hyödyntämiselle. Luvun lopussa kerrotaan käytön laajentamisen mahdollisuuksista ja heikkouksista.

4.1 Toiminta-alueet, joilla kohdeyritys hyödyntää Tekla Structures -ohjelmistoa

Kohdeyrityksessä käytännössä kaikki hankkeet tietomallinnetaan käyttäen Tekla Structures ohjelmistoa. Tietomalleja hyödynnetään myös hankkeissa, joissa kohdeyritys toimii rakennuttajakonsulttina, jolloin tietomalleja voidaan hyödyntää esimerkiksi valvonnassa ja määrälaskennassa. Tässä työssä ei kuitenkaan rajallisten resurssien vuoksi keskitytty näiden työntekijöiden tarpeiden selvitykseen. Tulevaisuudessa selvitystyötä suositellaan tehtäväksi myös muiden käyttäjien kuin rakennesuunnittelijoiden keskuudessa.

Tällä hetkellä Tekla Structures Open API sovelluksissa keskitytään ylläpitämään pakollisia, kuten esimerkiksi nimiöasetuksia ja olemassa olevia työkaluja.

4.2 Tavoitteiden selvitys

Nykytilannetta ja tarpeita selvitettiin haastatteleamalla BIM manager Teemu Ahosta sekä tekemällä kyselytutkimus Teklan käyttäjille. Vapaamuotoisten keskusteluiden pohjalta pyrittiin saamaan karkea käsitys siitä, mitä avoimen rajapinnan avulla on mahdollista tehdä ja minkälaisen ongelmien ratkomiseen sitä voi käyttää. Kyselytutkimuksella laajennettiin näkemystä ja kartoitettiin, mitkä ongelmat vaikuttavat rakennesuunnittelijoiden

päivittäiseen työhön ja minkä käyttäjät kokevat tärkeäksi. Tähän kappaleeseen on koottu olennaisimmat ajatukset kehitystyön jatkamisesta yrityksessä.

4.2.1 Keskustelu BIM managerin kanssa

Tarpeiden kartoitus aloitettiin vapaamuotoisilla keskusteluilla kohdeyrityksen BIM manager Teemu Ahosen kanssa. Keskusteluiden pohjalta tärkeiksi asioiksi nousivat ensin selvittää Open API:n tarjoamat erilaiset mahdollisuudet ja niiden keskinäiset erot kehitystyön kannalta. Tavoitteiksi asetettiin tutkia mitä erilaisia pieniä työkaluja voidaan kehittää, joiden avulla voidaan helpottaa päivittäistä työtä. Yhdessä tällaiseksi esimerkiksi nousi työkalu, joka helpottaa paaluluettelon laatimista tulostamalla paalujen korkopisteet tietomallista aina oikein, nykyisellään korkopiste riippuu mallintamissuunnasta ja siksi korko ei aina ole oikein, mikäli kyseessä on esimerkiksi vinoon lyötävä paalu.

Tärkeäksi nousi myös tietojen tulostaminen tietomallista myöhempää käyttöä varten. Tietomallin sisältämä tieto ei ole nykyisellään tarpeeksi helposti saatavilla ja tiedon siirtäminen vaatii turhaa toistuvaa työtä, joka olisi verrattain helppo automatisoida. Yksi esimerkki tällaisesta on valmiiden piirustusten siirtäminen projektipankkiin. Tällä hetkellä kuvat siirretään yksitellen ja myös kuvien tiedot aina projektinumerosta tekijöiden tietoihin siirretään käsin. Open API mahdollistaisi kaikkien näiden tietojen siirtämisen kerralla, jolloin työntekijöiltä vapautuisi aikaa itse suunnittelutyöhön.

4.2.2 Kyselytutkimus käyttäjille

Kyselytutkimus Tekla Structures -ohjelmiston käyttäjille toteutettiin kohdeyrityksen Tampereen rakennesuunnittelun yksikössä, jossa työskentelee noin 20 rakennesuunnittelijaa. Suunnittelijoiden ikä ja työkokemus alalta vaihtelee ja näin ollen yksikkö tarjoaa kattavan läpileikkauksen koko kohdeyrityksen rakennesuunnittelijoista. Yksikkö suunnittelee teollisuuskohteita ja kohteiden joukosta löytyy sekä teräs- että betonikohteita ja vaihtelevasti myös puukohteita. Mukana oli vain pääasiallisesti suunnittelutyötä tekeviä henkilöitä, eli esimerkiksi kehitystyöhön erikoistuneiden vastauksia ei ole mukana. Tarkoitus oli kerätä tietoa rivikäyttäjiltä, eikä ominaisuuksiin perehtyneiltä työntekijöiltä. Kysely toteutettiin käyttäen Google Forms palvelua ja vastaukset kerättiin nimettöminä. Aikaa vastaamiseen annettiin kaksi viikkoa. Kysymykset jaettiin kahteen osaan, ensin kyseltiin taustoja mallinnusohjelmien käytöstä, jonka jälkeen on Tekla Structures -ohjelmiston käytettävyyteen ja omaan työskentelyyn liittyviä kysymyksiä. Kysymykset on lueteltu alla osioittain.

Kysymysten tarkoitus on kartoittaa ensin käyttäjien taustat ohjelmiston parissa ja sen jälkeen saada vastauksia tutkimuskysymyksiin. Kysymysten joukossa on kolme monivalintakysymystä, kysymykset yksi, kaksi ja viisi, kaksi kyllä tai ei -kysymystä, kysymykset 3 ja 4 ja kahdeksan avoimen vastauksen kysymystä. Vastaajien ei ole pakko vastata kaikkiin kysymyksiin. Vastaukset kerättiin nimettöminä. Kyselytutkimuksen kysymykset on listattu alla:

- Taustakysymykset
 1. Kauanko olet käyttänyt Tekla Structures -ohjelmistoa?
 2. Kuinka usein tarvitset työssäsi Tekla Structures -ohjelmistoa?
 3. Olen käyttänyt muita mallinnusohjelmia?
 4. Tunnistatko käsitteen Open Api?
- Ohjelmistoon liittyvät kysymykset
 5. Anna arvosana Tekla Structures -ohjelmiston käytettävyydelle. 1 huonoin, 5 paras
 6. Mikä tai millainen työkalu tehostaisi mielestäsi eniten työntekoasi?
 7. Mikä ärsyttää sinua eniten käyttäessäsi Tekla Structures -ohjelmistoa?
 8. Onko Tekla Structures-ohjelmistossa ominaisuuksia, jotka turhauttavat sinua?
 9. Koetko, että käytät liikaa aikaa johonkin toistuvaan tehtävään, jonka automatisointi tehostaisi työtäsi merkittävästi? Jos kyllä, mihin?
 10. Mikä ärsyttää sinua eniten käyttäessäsi Tekla Structures -ohjelmistoa?
 11. Arvioi aika jonka kuukaudessa kuluu virhetilanteisiin tai ohjelmiston toimimattomuuteen.
 12. Olisitko itse kiinnostunut osallistumaan kehitystyöhön, mikäli se ei vaadi suurta tietoteknistä osaamista?
 13. Vapaa sana

Kyselyn kysymykset valittiin työn aikaisempien vaiheiden tutkimusten perusteella. Kysymykset pyrittiin koostamaan niin, että kyselyyn on mahdollisimman helppo ja mutkaton vastata ilman osaamista Tekla Structures Open API:n parissa. Ohjelmistoon liittyviin kysymyksiin jätettiin mahdollisimman paljon avoimuutta, jotta käyttäjillä oli mahdollisuus kertoa hyvin laajasti omia kokemuksiaan ja mielipiteitään näin halutessaan.

4.2.3 Kyselyn tulokset

Kyselyyn saatiin yhteensä 12 vastausta. Ensimmäiset viisi kysymystä kartoittavat vastaajien taustoja Tekla Structures -ohjelmiston parissa. Ensimmäisen taustakysymyksen perusteella saatiin selville, että kaikki vastaajat olivat käyttäneet Tekla Structures -ohjelmistoa vähintään vuoden, neljä henkilöä vähintään viisi vuotta. Vastaajista 10 käyttää ohjelmistoa päivittäin työssään ja kaksi vähintään kolmena päivänä viikossa. Muiden mallinnusohjelmien käyttö jakautui tasan puoliksi, kuusi on käyttänyt muitakin mallinnusohjelmia ja loput eivät. Käsite Open Api oli tuntematon suurimmalle osalle vastaajista, sillä kahdeksan vastaajaa ei käsitettä tunnistanut.

Ensimmäinen kysymys Tekla Structures -ohjelmiston kehittämisestä, eli kysymys numero viisi pyysi vastaajia arvioimaan ohjelmiston yleistä käytettävyyttä. Ohjelmiston käytettävyydelle yleisin arvosana oli neljä asteikolla yhdestä viiteen, arvosanan neljä antoi yhteensä kahdeksan vastaajaa, kolme vastaajaa antoi arvosanan kolme ja yksi antoi arvosanan viisi. Vastausten keskiarvoksi saatiin 3,83.

Avoimien kysymysten kaikki vastaukset on esitetty liitteessä A. Ensimmäisen avoimen kysymyksen, eli kysymyksen numero kuusi, vastausten perusteella eniten työntekoa tehostaisi työkalu, joka yhdistäisi rakenteiden mitoituksen osaksi tietomallinnusta, tämä esiintyi neljässä vastauksessa. Esimerkeiksi mainittiin liitosten kestävyys laskeminen geometrian perusteella ja Robot Structural Analysis ohjelmistolla. Kahdessa vastauksessa mainittiin piirustusten muokkaaminen tehokkaammin ja yhdessä vastauksessa älykkäämpi osien yhteensopivuus.

Ohjelmistoa käytettäessä suurinta ärsytystä kyselyyn vastanneissa käyttäjissä herättivät kysymyksen numero seitsemän vastausten perusteella piirustuspuolen ominaisuudet viidessä vastauksessa ja ohjelmiston hitaus, joka mainitaan neljässä vastauksessa. Ohjelmiston bugit tai yleinen toimimattomuus on mainittu kolmessa vastauksessa. Lisäksi mainittiin muutokset ohjelmistoversioiden välillä kolmessa vastauksessa ja erityisesti nostetiin esille tuttujen työkalujen muutokset.

Turhautumista ohjelmistossa koettiin samoissa ongelmissa kuin edellisen kysymyksen kohdalla ja se näkyy myös vastausten määrässä, joita saatiin vain seitsemän kappaletta kysymykseen numero kahdeksan. Piirustuspuolen ongelmat mainittiin kahdessa vastauksessa. Mallinnuksen manuaalisuus ja referenssitiedostojen ongelmat mainittiin kumpikin yhdessä vastauksessa. Lisäksi yhdessä vastauksista moititaan organizer -lisäosan toimimattomuutta. Organizer -lisäosa on kohdeyrityksen oman kehitystyön tulos, joka on herättänyt kiitosta käyttäjien keskuudessa yleisesti.

Piirustukset mainittiin kuudessa vastauksessa yhdeksästä kysyttäessä tehtäviä, joiden automatisointi tehostaisi työntekoa kysymyksessä numero yhdeksän. Neljässä vastauksessa mainittiin asetusten asettamisen tai kopioinnin hankaluus. Vastausten perusteella aikaa kuluu turhaan toistoon piirustusten teon yhteydessä useassa kohdassa. Yhdessä vastauksista esitettiin myös idea Rhinoceros 3D-ohjelman Grasshopper-lisäosan Tekla Live Lin-kin käyttöönotosta.

Kysymyksessä numero kymmenen kysyttäessä aikaa, joka kuluu kuukaudessa ohjelmiston virhetilanteisiin tai ohjelmiston toimimattomuuteen, saatiin kymmenen toisistaan eroavaa vastausta. Vastausten hajonta johtunee kysymyksen asettelusta, joka ei ollut yksiselitteinen. Vastauksissa aika-arvio vaihtelee 20 minuutista kahdeksaan tuntiin. Kun vastauksista jättää suurimman ja pienimmän arvion pois, osuvat kaikki muut vastaukset tunnin ja kahden tunnin välille. Ohjelmiston kehitystyö herätti kahdeksassa vastaajassa kahdestatoista kiinnostusta kysymyksessä yksitoista.

Viimeiseen vapaamuotoiseen kysymykseen numero kaksitoista saatiin yhteensä kuusi vastausta. Kahdessa vastauksessa kritisoitiin, sitä että ohjelmiston uusi versio ei tarkoita parempaa versiota ja versiot julkaistaan keskeneräisinä. Esiin tuotiin myös Grasshopper ja parametrinen mallintaminen, sekä oma käyttöliittymä mallin ja piirustusten operointiin.

Lisäksi useissa vastauksissa eri kysymysten kohdalla mainittiin ohjeiden ja manuaalien heikkoudet, jotka eivät suoraan liity ohjelmistokehitykseen.

4.2.4 Tulosten analyysi

Vastauksista käy ilmi, että Tekla Structures koetaan pääpiirteittäin hyväksi ohjelmaksi ja käyttäjät osaavat nimetä ongelmakohtia joihin puuttuminen tekisi työstä sujuvampaa. Vastauksista huomaa, että asiaa on mietitty ja ongelmien listaamisen lisäksi vastauksissa oli myös konkreettisia kehitysideoita ja tietoa kehitysmahdollisuuksista. Kehitystyö nähdään tärkeäksi ja sen mahdollisuuksiin uskotaan. Ohjelman ei koeta olevan huono ja ohjelma saa paljon kehuja, mutta silti kehitystarpeet ja mahdollisuudet tiedostetaan hyvin.

Kysyttäessä eniten työskentelyä tehostavaa työkalua kysymyksessä kuusi, vastaajat kokevat rakennesuunnittelijan työn jakautumisen eri ohjelmistojen välille ongelmaksi. Nykyisellään rakenteet ja liitokset mitoitetaan ensin ja mallinnetaan vasta tämän jälkeen ja mikäli mitoitusta muutetaan, on sen jälkeen tarvittaessa muutettava myös mallia. Työvaiheiden yhdistäminen ohjelmistojen rajapintojen kautta ja näin turhan toiston välttäminen koetaan merkitykselliseksi. Osa käyttäjistä on vastausten perusteella tietoisia siitä, että avoimien rajapintojen kautta erilaisia suunnitteluohjelmistoja on mahdollista yhdis-

tää ja tätä kaivataan. Asia on jatkuvasti pinnalla ja se osaltaan selittää, miksi työn jakautuminen eri ohjelmistoihin koetaan turhauttavammaksi kuin esimerkiksi piirustuksiin liittyvät ongelmat.

Vastauksissa piirustukset nousevat selkeäksi kehityskohdaksi, vaikka niitä ei mainita kuin kahdessa vastauksessa kysymyksessä kuusi. Piirustusten teon hitaus ja turhaan toistoon turhautuminen esiintyy lähes jokaisen kysymyksen vastauksissa. Piirustuksia tehdään paljon etenkin teräsrakenteita suunnitellessa ja niiden muokkaamiseen käytetään paljon aikaa, jolloin piirustusten tekemisen tehostaminen on olennaista. Piirustusten määristä ja niihin käytetystä ajasta huolimatta vastaajat eivät koe, että piirustusten teon tehostaminen ja työkalun kehitys tehostaisi heidän työskentelyään eniten.

Useissa vastauksissa toivotaan joko parempia käyttöohjeita tai videoita mallintamisesta ja uusien versioiden käyttöönotto koetaan raskaaksi samoin kuin työkalujen muuttuminen. Uusien ohjelmistoversioiden koetaan olevan säännöllisesti edeltäjiänsä huonompia ja aiheuttavan lisäkuormitusta työhön. Puutteellinen osaaminen ja tunne omasta osaamattomuudesta näkyvät joidenkin vastausten kohdalla tyytymättömyytenä ohjelmistoon. Tekla Structures on kaupallinen ohjelmisto, jonka uusien versioiden julkaisutahtiin tai julkaisuiden sisältöön kohdeyritys ei voi mitenkään vaikuttaa, mutta se korostui silti käyttäjien vastauksissa. Tähän ongelmaan ei valitettavasti voida avoimen rajapinnan kehitystyön puitteissa kehittää ratkaisua, mutta käyttäjien tukea ulkoisten muutosten edessä voidaan kehittää ja tietoisuutta lisätä tarvittaessa.

Tutkimuksen otos oli yrityksen kokoon nähden pieni ja tutkimukseen vastanneet käyttäjät kuuluivat kaikki samaan yksikköön, joten laajemmalla kyselyllä olisi saanut kattavamman kuvan nykytilasta ja mahdollisesti täysin erilaisiakin ongelmia tai vahvuuksia olisi noussut esiin. Parhaan käsityksen hukkaan menneestä työajasta mallinnusohjelmiston parissa saisi seuraamalla työntekijöiden työskentelyä esimerkiksi seuraamalla virheilmoituksia tai käyttäjien aktiivisuutta töitä tehdessä. Mahdollisuuksia laajemman selvityksen tekemiseksi löytyy lähes rajattomasti, mikäli tulevaisuudessa kaivataan tarkempaa dataa käyttäjien tarpeista.

4.3 Avointen rajapintojen käyttöön liittyvät keskeiset haasteet ja mahdollisuudet

Sekä keskusteluiden, että haastatteluiden perusteella voidaan todeta, että Tekla Structuresin kehitystyö Open API:n kautta on tarpeen ja se koetaan mahdollisuutena. Kummassakin käy ilmi, että esimerkiksi muiden ohjelmistojen toimiminen yhteistyössä Teklan kanssa koetaan tärkeäksi. Kummastakin näkökulmasta piirustukset nousevat vahvasti esille.

Asiantuntija ja kehitystyön tunteva henkilö näkee tarpeet hyvin eri kohdissa kuin rivi-käyttäjät ja tässä on syytä huomioida, että kehitystyötä tehdään käyttäjille, mutta kuitenkin muistaen, että kysely oli siinä mielessä vajavainen, että käyttäjille ei kerrottu Open API:n tuomista mahdollisuuksista, joten on hyvin mahdollista, etteivät käyttäjät tiedä kaikkien mahdollisuuksien olemassaolosta ja siksi osaa nimetä niitä.

On tärkeää varmistaa tarvittavat resurssit ennen kuin kehitystyöhön ryhdytään ja luotava selkeä suunnitelma, jonka mukaan kehitystyötä tehdään. On tärkeää, ettei käyttäjiä kuormiteta turhilla muutoksilla ja uudistuksilla, on myös tuettava uusien sovellusten ja työkalujen käyttöönottoa. Puolivalmiit työkalut eivät ole toivottuja ja työkalujen käytettävyys on testattava ennen käyttöönottoa ja panostettava myös ylläpitoon. Suunnittelijat turhautuvat herkästi, mikäli uudistuksien jälkeen työskentely ei ole helpompaa. Hyväkin työkalu menee hukkaan, jos sitä ei haluta käyttää puutteellisten ohjeiden tai epävakaa toiminnan takia.

Työkalut, joita kehittäjä haluaa, eivät suoraan välttämättä helpota mallintamista vaan enemmänkin tietomallin hyödyntämistä. Käyttäjät keskittyvät tutkimuksen perusteella enemmän ongelmiin mallin ja piirustusten tekemisen aikana. Tämä on huomioitava pohdittaessa tärkeimpiä kehityskohteita.

Selkeä suunnitelma Open API:n hyödyntämisestä ja vaadittavasta osaamisesta sen toteuttamiseksi on tärkeä, jotta olemassa olevat resurssit kohdistetaan oikein. Trimble tarjoaa kattavan kirjaston pohjaksi kehitystyölle ja yhteisön kehitystyön tueksi, joten olemassa olevan materiaalin hyödyntäminen on muistettava. Parametrinen mallintaminen on kasvava trendi rakennusosalalla ja sen käyttö yleistyy ja myös sen tuomat hyödyt on otettava tarkasteluun suunniteltaessa kehitystyötä.

5. OPEN API -SOVELLUSTEN KÄYTÖN LAAJENTAMINEN JA TEHOSTAMINEN KOHDEYRITYKSESSÄ

Tässä kappaleessa annetaan ehdotuksia open API sovellusten käytön laajentamiseksi kohdeyrityksessä. Neuvoja annetaan kehitystyön tueksi ja kehitystyön kompastuskivien välttämiseksi, sekä konkreettisia ehdotuksia uusista työkaluista. Ohjeet käytön laajentamiseen on jaettu kahteen osaan, teknisiin ideoihin käytön laajentamiseksi ja ajatuksiin ja ehdotuksiin työkaluista, joilla tuetaan käytön laajentamisen onnistumista, huolehtien käyttäjien tyytyväisyydestä ja resurssien onnistuneesta käytöstä niin, että saavutetaan mahdollisimman suuri hyöty suhteessa käytettyihin resursseihin. Näiden kahden osa-alueen lisäksi tässä kappaleessa perehdytään parametrin mallintamisen periaatteen, taustoihin ja mahdollisuuksiin yrityksessä.

5.1 Kehitystyössä huomioitavaa

Tässä kappaleessa kerrotaan, kuinka yritys parhaiten ottaa huomioon käyttäjien tarpeet kehitystyössä ja välttää yleisimmät virheet. Kehitystyön mielekkyyden ja parhaan lopputuloksen varmistamiseksi käyttäjät on otettava huomioon heti alusta asti, mutta kuitenkin unohtamatta heitä myöskään kehitystyön edetessä. Tekla Open API:n materiaaleissa ja manuaaleissa kerrotaan, että avoimen rajapinnan hyödyntämistä ei suositella tuotavaksi kaikkien suunnittelijoiden työkaluksi, vaan yrityksen tulee nimetä henkilöt, joilla on riittävästi osaamista mallintamisesta ja joilta löytyy kiinnostusta kehitystyötä ja ohjelmointia kohtaan tai jopa osaaminen kumpaankin entuudestaan. (Trimble Solutions Corporation, 2018)

Trimble ohjeistaa myös aloittamaan avoimen rajapinnan käytön tavoitteiden asettamisella pitkälle lyhyelle tähtäimelle niin, että olemassa olevat resurssit mahdollistavat tavoitteiden täyttämisen. Tämä pätee myös, kun yritys haluaa laajentaa avoimen rajapinnan hyödyntämistä merkittävästi. Tavoitteet, jotka ovat epärealistisia ja mahdottomia toteuttaa aiheuttavat turhautumista ja epäuskoa. Kehitystyö suositellaan aloitettavaksi ratkaisemalla suurimmat suunnittelun pullonkaulat yksi osa-alue kerrallaan. (Trimble Solutions Corporation, 2018) On mielekkäämpää tuottaa vähemmän toimivaa ja hyödyllistä sisältöä kuin paljon toimintavarmuudeltaan heikkoa ja mahdollisesti jopa tarpeetonta tai haitallista sisältöä.

Kehitystyön kannattavuuden ja pitkäjänteisyyden varmistamiseksi dokumentointi on otettava huomioon alusta alkaen ja sille on luotava selkeät pelisäännöt, näin esimerkiksi työkalujen jatkojalostaminen ja ylläpitäminen on helpompaa, aina muistaen tarkistaa, ettei ongelmaan löydy jo valmista sovellettavaa ratkaisua. Käyttäjät on syytä osallistaa kehitystyöhön. (Trimble Solutions Corporation,)

Työvälineet ovat rakennesuunnittelijoille päivittäisessä käytössä ja toimivilla työkaluilla on suuri merkitys siihen, millaiseksi työn tekeminen muodostuu. Työhyvinvointi syntyy työn arjessa ja on investointi, joka parhaimmillaan vaikuttaa organisaation kilpailukykyyn, taloudelliseen tulokseen ja maineeseen, joka työpaikasta syntyy, jolloin investoinnit parhaimmillaan maksavat itsensä takaisin. Hyvällä työhyvinvoinnilla on tutkimusten mukaan selkeä vaikutus mm. työntekijöiden vähäisempään vaihtuvuuteen ja sairauspoissaolojen määrään. (Työhyvinvointi.)

Työn kehittämisen merkitys korostuu, kun työelämä on jatkuvassa murroksessa. Kehittäminen on syytä tehdä yhteistyönä, johon osallistuvat työntekijät, esimiehet ja kehittämisasiantuntijat yli asiantuntija- ja organisaatorajojen. Paras tulos saavutetaan, kun kasvatetaan aktiivista toimijuutta ja otetaan koko yhteisö mukaan kehitystyöhön ja annetaan jokaiselle mahdollisuus vaikuttaa oman työnsä mielekkyyteen. (Työn kehittäminen.)

Osana diplomityötä tutustuin ohjelmointiin ja avoimiin rajapintoihin. Työn alkuperäisenä ajatuksena oli tuottaa yritykselle toimivia työkaluja hyödyntäen Teklan avointa rajapintaa, mutta tämä ajatus osoittautui liian kunnianhimoiseksi, sillä vaikka kehitysympäristöä kehitetään helppo käyttöiseksi ja materiaalien mukaan Tekla Open API:n referenssikirjastojen ja valmiiden työkalujen avulla lisäosien rakentaminen on helppoa, vaativat jo harjoitustehtävien ensimmäisetkin osat ohjelmoinnin opiskelua.

Seuraavissa kappaleissa syvennytään kehitystyön laajentamisen ja sen onnistumisen kannalta kolmeen yritykselle merkilliseen asiaan, perehdytykseen, jatkuvan palautteen keräämiseen ja käyttäjien osallistamiseen sekä avoimen rajapinnan avoimuuden toteuttamiseen myös yrityksen sisällä.

5.1.1 Perehdytys

Jenni Tossavainen tutkii marraskuussa 2006 julkaistussa Pro Gradu työssään ”Työhön perehdytys asiantuntijaorganisaatiossa” perehdytyksen toimivuutta ja merkitystä työpaikan tulokkaan kehittämisessä asiantuntijaksi. Perehdytyksen onnistumiseen vaikuttavat tulokkaan aiempi työkokemus, oma aktiivisuus ja kontaktihakuisuus. Perehdytyksen jälkeen erityisen suuri merkitys on osaamisen lisäksi tiedonhaulla ja sosiaalistumisella. Hy-

vä perehdytys auttaa tulokasta pääsemään osaksi työyhteisöä heti työsuhteen alussa. Asiantuntijuus kasvaa työyhteisössä ja tiedonhaun onnistumiseen avainasemassa ovat samoissa projekteissa työskentelevät kollegat, erityisesti työsuhteen alussa mielellään tietoaan ja osaamistaan jakavat kollegat ovat merkittävässä roolissa. (Tossavainen, 2006)

Tossavainen antaa tutkimustuloksissaan suosituksen siitä, että perehdytys tulisi organisaatiossa jakaa kahteen osaan, yleiseen ja yksilölliseen osuuteen. Yleisessä osuudessa käydään läpi yleiset asiat työyhteisöstä ja yrityksestä ja tutustutaan työtovereihin. Henkilökohtaisessa perehdytyksessä huomioidaan tulokkaan omat taustat ja opastetaan tarvittavien tietokoneohjelmien käyttöön ja työyhteisön rutiineihin ja määritellään tavoitteet. Varmimmiksi tavoiksi epäonnistua perehdytyksessä Tossavainen nimeää epäpätevän perehdyttäjän ja liiallisen kiireen perehdytyksessä. (Tossavainen, 2006)

Yrityksen henkilöstölle teetetyssä kyselytutkimuksessa havaittiin kehityskohteita perehdytyksessä ja on suositeltavaa, että yrityksessä otetaan käyttöön Tossavaisen suosittama kaksivaiheinen perehdytys ja varmistetaan, että perehdytyksen antajalla on tarvittavan tietotaidon lisäksi tarpeellinen määrä aikaa laadukkaaseen perehdytykseen. Tämä voi olla haastavaa työn projektiluonteisuuden vuoksi, mutta perehdytyksellä vaikutetaan ratkaisevasti tulokkaan asiantuntijuuden kehittymiseen, joten laadukkaalla perehdytyksellä saadaan myös tulokas mahdollisimman nopeasti auttamaan kiireisissä projekteissa. Työturvallisuuskeskuksen julkaisemassa oppaassa ”Työhön perehdyttäminen ja opastus – ennakoivaa työsuojelua” myös muistutetaan, että perehdytys voidaan nähdä investointina optimaaliseen työntekijään. Perehdyttämisellä kasvatetaan henkilöstön osaamista ja tuetaan asiantuntijoiden työssä jaksamista ja näin myös parannetaan työn laatua. (Penttinen & Mäntynen, 2009)

Yksilöllisessä perehdytyksessä tehdään jokaiselle tulokkaalle henkilökohtainen perehdytys suunnitelma (Tossavainen, 2006). Suunnitelmassa otetaan huomioon tulokkaan osaamistaso Tekla Structuresin käytössä ja varmistetaan, että jokaisella on tarpeeksi aikaa perehtyä ohjelmiston käyttöön ennen varsinaisia projektitöitä. Tossavainen nostaa tutkimuksessaan myös esiin sen, että aivan liian usein rutiinikomennot ja rutiinitehtävät ovat kokeneille tekijöille jo itsestäänselvyksiä, voivat jäädä kertomatta tulokkaalle ja aiheuttaa merkittävää hidastumista ja turhautumista työskentelyyn (Tossavainen, 2006). Perehdytyksistä on syytä kerätä palautetta kaikilta tulokkailta, jotta havaitaan mitä perehdytyksessä on syytä kehittää ja mikä on onnistunut. Toisaalta palautetta kerätessä voidaan varmistaa myös se, että tulokas kokee pärjäävänsä ohjelmistojen ja työtehtävien

kanssa ja tarvittaessa voidaan jo pian perehdytyksen jälkeen tehdä korjausliikkeitä, mikäli perehdytys on joiltain osin jäänyt vajaaksi. Perehdytys on parhaimmillaan kehittyjä ja jatkuva prosessi (Penttinen & Mäntynen, 2009).

Tekla Structures ohjelmistosta on saatavilla sekä Trimblen, että kokeneiden käyttäjien tekemiä opetusvideoita ja ohjeita ja näitä voidaan käyttää perehdyttämisen tukena, mutta tulokkaalla on oltava myös neuvonantajia, joilta kysellä ohjeita ja vinkkejä. Neuvonantajien tulee suhtautua tulokkaan kysymyksiin rakentavasti ja kyselyvaiheelle on varattava tarpeeksi aikaa, jotta kysyminen ja asioiden selvittäminen on tulokkaalle mielekästä. Tarvittaessa tulokkaalle tarjotaan myös ohjattua opastusta ohjelmiston saloihin, joko heti työsuhteen alussa tai myöhemmin perehdytyksen jälkeen. Mikäli huomataan myöhemmin, että henkilökohtainen arvio tulokkaan osaamisesta on syystä tai toisesta ollut virheellinen, tarjotaan myös myöhemmin mahdollisuutta lisäkoulutukseen.

Tossavaisen tutkimuksessa selviää myös, että oman työn kehittämistä pidetään tärkeänä (Tossavainen, 2006). Kertomalla jo perehdytysvaiheessa tulokkaille Tekla Structuresin avoimista rajapinnoista ja niiden parissa tehtävästä kehitystyöstä ja työkalujen palautekanavista kerrotaan tulokkaalle yhdestä tavasta vaikuttaa omaan työhönsä.

5.1.2 Jatkuva palaute ja käyttäjien osallistaminen

Tekla Structuresin käyttäjille tehdyssä kyselytutkimuksessa havaittiin, että osa vastaajista ei koe kaikkia ohjelmiston uudistuksia ja uusia työkaluja välttämättä mielekkäiksi käyttää tai työtä helpottaviksi. Tähän ongelmaan yritys voi pureutua kahdella tapaa, uusien työkalujen koulutuksilla ja opastuksilla, sekä versiopäivitysten tietoisuilla, joilla varmistetaan, että työntekijät saavat tarkoitetut hyödyt irti eivätkä päivitykset ja uudet työkalut rasita työntekijöitä, sillä se ei ole tarkoituksenmukaista. Työntekijöiltä kerätään myös jatkuvaa palautetta uudistuksista ja palautteen anto kanavat rakennetaan niin, että palautteen antaminen koetaan tehokkaaksi ja palautteisiin reagoidaan tehokkaasti ja palautteen antaja saa aina tiedon siitä, että palaute on vähintäänkin päätynyt käsiteltäväksi, mikäli se ei päädy aiheuttamaan toimenpiteitä. Palautteen käsitteleminen läpinäkyvästi lisää työntekijän kokemusta siitä, että hänellä on aito mahdollisuus vaikuttaa omaan työhönsä.

Kehitystyötä tehdään käyttäjille eikä kehittäjille. Tämä on syytä muistaa joka kerta kun pohditaan strategiaa ja suuntauksia kehitystyölle. Kehittäjille mielenkiintoisimmat projektit ja toteutukset eivät välttämättä palvele käyttäjien tarpeita ja näissä tilanteissa on priorisoitava käyttäjien mielipiteet. Tämän mahdollistamiseksi on palautettava kerättävä säännöllisesti ja pelkkien arvostelu palautteiden lisäksi on jatkuvasti kerättä käyttäjiltä

kehitysideoita ja parannus ehdotuksia ja näiden palautteiden läpikäymiseen on varattava tarpeeksi resursseja muun kehitystyön lomaan. Tehokkailla ja toimivilla palautekanavilla käyttäjät otetaan tiiviisti mukaan kehitystyöhön.

Kehitystyötä tehtäessä on pohdittava uusien työkalujen ja sovellusten käyttäjäkokemusta. Käyttäjäkokemuksella tarkoitetaan ihmisen ja tässä tapauksessa ohjelmiston välistä vuorovaikutusta. Käyttäjäkokemus voi liittyä myös esimerkiksi esineiden tai muiden tuotteiden ja ihmisen väliseen vuorovaikutukseen. Käyttäjäkokemus määritellään henkilön havainnoiksi ja reaktioiksi, jotka ovat seurausta tuotteen käytöstä tai sen ennakoidusta käytöstä (Tryfonas, 2016). Käyttäjäkokemus voi määritelmän mukaisesti olla huono, vaikka käyttäjä ei vielä olisi esimerkiksi ohjelmistoa käyttänyt, mikäli edeltävien kokemusten perusteella käyttäjällä on syytä olettaa, että ohjelmisto ei ole mieleinen.

Käyttäjäkokemus on subjektiivinen tuntemus, ja keskeisenä tekijänä käyttäjäkokemuksen syntyyn on tuotteen, eli tässä tapauksessa ohjelmiston käytettävyyys. Käytettävyyden merkitys on läsnä kaikessa suunnittelussa, esineiden ja tuotteiden on pohjimmiltaan tarkoitus toimia hyvin ja tuottaa käyttäjilleen mielihyvää. Paras mahdollinen käytettävyyys varmistetaan toteuttamalla kehitystyössä käyttäjäkeskeistä suunnittelua. Käyttäjäkeskeinen suunnittelu tarkoittaa prosessia, jolla tuotteeseen tai järjestelmään suunnitellaan käytettävyyttä. Tavoitteena on käyttäjäkeskeisen suunnittelun kautta tuoda suunnittelu-prosessiin tietoa käyttäjiltä ja perustaa suunnitteluratkaisut käyttäjätietoon eikä, suunnittelijoiden intuitioon. Käyttäjäkeskeisen suunnittelun prosessi voidaan tiivistää neljään osaan: käyttäjien aktiivinen osallistaminen, toimintojen kohdentaminen käyttäjien ja teknologian välillä, suunnitteluratkaisujen iterointi ja monialainen suunnittelu. (Väänänen, 2016)

Yrityksessä käyttäjäkeskeistä suunnittelua voidaan soveltaa yrityksessä jo olemassa olevaa Tekla Structures ohjelmiston pääkäyttäjien verkostoa sekä rivisuunnittelijoita, jolloin saadaan kattavampi otos käyttäjistä kuin satunnaisella otannalla. Avoimen rajapinnan kautta on mahdollista kehittää työkaluja tekemään lähes mitä tahansa käyttäjien puolesta, mutta käyttäjäkeskeisen suunnittelun periaatteiden mukaan annetaan käyttäjien määritellä itse, mitkä tehtävät he haluavat tehdä itse ja mihin kaipaavat helpotusta. Suunnitteluratkaisujen iterointia voidaan yrityksessä soveltaa esimerkiksi niin, että uudet työkalut tuodaan ensin suppeamman käyttäjäkunnan kokeiltavaksi, aloitetaan käyttäjistä, joilla on mielenkiintoa osallistua kehitystyöhön ja heidän kokemustensa perusteella kehitetään työkalua edelleen ennen sen julkaisua jälleen suuremmalle yleisölle. Näin valmis ja koko käyttäjäkunnan käyttöön päätyvä sovellus tai työkalu on läpikäynyt jo useita iteraatiokierroksia, joilla on korjattu mahdollisia huonon käytettävyyden ratkaisuja.

5.1.3 Avoimen rajapinnan sisäinen avoimuus

Trimble ohjeistaa, että avointa rajapintaa ei ole tarkoitettu kaikkien suunnittelijoiden käyttöön, vaan suurin hyöty saadaan, kun työkaluja kehittää erillinen tietotekniikkaan ja ohjelmistokehitykseen perehtynyt ryhmä (Trimble Solutions Corporation, 2018). Yrityksen ei kannata velvoittaa kaikkia Tekla Structuresin käyttäjiä perehtymään avoimen rajapinnan toimintaan ja sen mahdollisuuksiin, mutta kuten luvussa kaksi on todettu, avoimien rajapintojen suurimmat hyödyt ovat siinä, että kuka tahansa pääsee kehittämään omia innovaatiotaan ja luomaan omia parempia ratkaisujaan. Myöskin edellisten kappaleiden neuvojen toteuttamiseksi kaikki halukkaat käyttäjät on syytä osallistaa mukaan kehitystyöhön.

Kansainväliset yritykset työskentelevät jo nyt erilaisten verkkolevyjen ja pilvipalveluiden avulla ja siksi tiedon jakaminen yrityksen sisällä on vaivatonta, eikä vaadi erityisiä resursseja. Avoimen rajapinnan kehitystyöhön liittyvät asiat on syytä koostaa jonnekin niin, että joko kaikilla käyttäjillä on suoraan pääsy tietoihin tai halukkaat voivat pääsyn pyytää. Näin tarjotaan yksi väylä lisää vaikuttaa omaan työhön ja samalla hyödynnetään koko yrityksen omien työntekijöiden potentiaali kehitystyössä. Tämä ei tarkoita, etteikö kehitystyöhön olisi oltava erikseen nimetty yksikkö tai ryhmä, mutta näin myös rivikäyttäjille tarjotaan mahdollisuus osallistua kehitystyöhön. Avoimen rajapinnan sovellusten ja mahdollisuuksien avaaminen saattaa myös herättää mielenkiinnon antaa enemmän ja rakentavampaa palautetta ja osallistua kehitystyöhön ideoinnin kautta, kun kehitystyö todella on avointa.

5.2 Piirustukset

Piirustukset nousivat käyttäjien keskuudessa esiin, kun kysyttiin, mitkä työtehtävät tuntuvat erityisen kuormittavilta. Myös Teklan omissa ohjeissa Tekla Open API:n hyödyntämisestä yrityksessä nimetään piirustusten teon tehostaminen ja pienten tehtävien automatisointi hyväksi ja helpoksi tavaksi aloittaa hyödyntäminen. Tekla Open API tarjoaa piirustus rajapinnan, jonka avulla voidaan helpottaa ja automatisoida piirustusten luomiseen ja muokkaamiseen liittyviä työtehtäviä. Rajapinnan avulla voidaan luoda kokonaan uusia piirustusobjekteja mukauttaa grafiikan esitystapaa, sekä muuttaa osien ja merkin-
töjen ulkoasua. Kuten muissakin Teklan rajapinnoissa, myös piirustusrajapinnassa voidaan käyttää kaikkia kolmea eri laajennustyyppiä. (Trimble Solutions Corporation, 2018)

Piirustusten teon helpottamiseksi Teklan harjoituskalvot suosittelevat erityisesti erilaisia helppokäyttöisiä makroja. Makrot ovat helppokäyttöisiä erityisesti, kun niiden tehtävät

ovat pieniä, eikä graafisen käyttöliittymän puute haittaa. Kalvoissa on annettu esimerkkejä kätevästä makroista, joita ovat esimerkiksi:

- Makro, joka poistaa muutoksia merkkäavat pilvikuviot.
- Makro, joka avaa piirustukset uudelleen ja pakottaa päivitykset.
- Makro, joka luo raudoituksen tai pulttiryhmän mitoituksen.

Makrojen lisäksi piirustus rajapinnassa voidaan työstää erillisiä piirustus applikaatiota, jotka toimivat Tekla Structuresin ulkopuolella ja kommunikoivat mallin ja piirustuksien välillä. Applikaatioiden avulla voidaan toteuttaa käyttöliittymää vaativat, makroja monimutkaisemmat kokonaisuudet. (Trimble Solutions Corporation, 2018)

5.3 Parametrinen mallintaminen

Käyttäjille suunnatussa kyselyssä parametrinen mallinnus nousi esiin useammassa vastauksessa. Yhdessä vastauksessa mainittiin parametrinen mallinnus ja Tekla live link suoraan ja useissa vastauksissa kaivattiin yksinkertaisten toimintojen automatisointia. Algoritmisen suunnittelun avulla jokin osa suunnittelua toteutetaan algoritmisen prosessin tuloksena perinteisen manuaalisen suunnittelun sijaan. Parametrinen mallinnus, eli tässä tapauksessa algoritmiaivusteinen suunnittelu soveltuu säännönmukaisiin toistuviin tehtäviin

Algoritmi on suoritettava komentosarja. Komentosarjaan määritetään rakenteeseen vaikuttavat säännöt. Algoritmiin syötetään parametrejä. Parametrejä muokkaamalla algoritmin tulos muuttuu ja näin voidaan reaaliajassa muokata mallia. Suunnittelun lopputuloksena on perinteisesti yksi malli, mutta algoritmiaivusteisen suunnittelun tuloksena on dynaaminen helposti muokattavissa oleva suunnitelma, mikä mahdollistaa esimerkiksi vaihtoman erilaisten vaihtoehtojen tarkastelun. (Tanska & Österlund, 2014)

Algoritmiaivusteinen suunnittelu onnistuu skriptikielillä osaavalta suunnittelijalta. Esimerkiksi Tekla Open API:n kautta C# kielellä. Ohjelmointikielten opiskelu ei ole kuitenkaan välttämätöntä, sillä suunnittelua varten on saatavilla myös plugin visuaaliseen skriptaukseen, tässä tapauksessa McNeel Grasshopper Rhinoceros 3D plugin. (Tanska & Österlund, 2014) Kuten aiemmin tässä kappaleessa on kerrottu, on ohjelmointi Tekla Open API:n avustuksella haasteellista ja työlästä mikäli aiempaa ohjelmointikokemusta ja -osaamista ei ole. Valmis työkalu, jolla voidaan välttää ohjelmointi, tehostaa siksi algoritmiaivusteisen suunnittelun käyttöönottoa.

Kyselytutkimuksen perusteella käyttäjillä on mielenkiintoa tutustua parametriseen mallintamiseen. Parametrinen mallintaminen on kiistatta nykypäivää rakennesuunnittelussa

ja siksi yhtenä osana rakennesuunnittelijoiden työn tehostamista mallinnusohjelman kehittämisen kautta, on huomioitava myös parametrinen mallintamisen mahdollisuudet.

5.3.1 Grasshopper

Grasshopper on ilmaiseksi saatava lisäosa Rhinolle. Grasshopperissa ei kirjoiteta tekstimuotoista ohjelmakoodia, eli skriptiä, vaan sama asia voidaan toteuttaa linkittämällä keskenään valmiita ohjelmakomponentteja. Visuaalisuus tekee Grasshopperin käytöstä helpommin lähestyttävää kuin skriptien kirjoittamisesta. (Tanska & Österlund, 2014)

Rhino on mallinnusohjelma, jota on helppo käyttää ja siksi myös suosittu. Se käyttää sisäisenä skriptikielensä RhinoScriptiä, joka on VisualBasic ohjelmointikieleen perustuva kuitenkin yksinkertaistettu mallinnusohjelman komentoihin perustuva kieli. Se on kuitenkin väistymässä Python-kielen kasvattaessa suosiota. Rhino on helppo räätälöidä kulloisiinkin tarpeisiin lisäosien avulla ja se soveltuu sekä tietomallintamiseen, että rakenneanalyysiin ja myöskin parametriseen mallintamiseen. (Tanska & Österlund, 2014)

Rhino pohjautuu NURBS (Non-Uniform Rational Basis Spline) -teknologiaan. NURBS-mallit ovat geometrialtaan tarkkoja matemaattisia esitystapoja erilaisille muodoille ja pinnoille aina yksinkertaisista viivoista moniulotteisiin pintoihin. NURBS-teknologiaan perustuvat mallit ovat aina matemaattisesti tarkkoja ja suurin osa mallinnusohjelmistoista ymmärtää niiden geometrian ja siksi teknologian avulla on helppo tallentaa geometrisia tietoja, joita voidaan käyttää myös jatkossa. (What are NURBS.2019)

Grasshopper on kohde yritykselle helppo tapa tutustua parametrinen mallintamisen mahdollisuuksiin ilman suuria investointeja ohjelmistoihin tai ohjelmointitaitoiseen henkilökuntaan.

5.3.2 Tekla live link

Tekla live link tarjoaa mahdollisuuden yhdistää Rhinon ja Grasshopperin mallinnusmahdollisuudet osaksi Tekla Structuresia. Live linkki on kokoelma Grasshopperin komponentteja, joilla käyttäjä voi reaaliaikaisesti luoda ja muokata komponentteja Tekla Structuresissa. Linkkiä voi käyttää kunhan samalle laitteelle on asennettu Rhino ja Tekla Structures, sekä vanhempia Rhino versioita (versio 5 tai vanhempi) käytettäessä myös Grasshopper, joka on sisällytettyä Rhino 6 asennuksessa. Grasshopper lisäosa on ilmainen, toisin kuin Rhino ja Tekla Structures. Asennusten jälkeen tarvittavat komponentit tulevat näkyviin Tekla-valikkoon Grasshopperissa. (Grillo & Krebs, 2018)

Tekla live link tarjoaa mahdollisuuden yhdistää Grasshopperin tarjoamat mahdollisuudet vielä vaivattomammin osaksi Tekla Structuresia kuin pelkkä Grasshopper. Mikäli yritys päätyy kokeilemaan parametrinen mallintamisen mahdollisuuksia Rhinon ja Grasshopperin avulla, Tekla live link ei tuo lisäkustannuksia. Kyselytutkimuksessa käyttäjillä oli kiinnostusta parametrista mallinnusta kohtaan ja Tekla live link tarjoaa helpon mahdollisuuden myös osallistaa käyttäjät kokeilemaan parametrista mallinnusta ilman, että investoidaan henkilökunnan koulutukseen, sillä Trimble tarjoaa hyvät ja selkeät ohjeet linkin käyttöön.

5.3.3 Tulevaisuuden näkymät

Parametrinen mallintaminen on viime vuosina ollut yksi rakennusalan megatrendeistä ja tuskin kukaan alalla työskentelevä on voinut välttää kuulemasta sen vahvuuksista. Kuten uusilla teknologioilla yleensä, myös parametrille mallintamiselle on löydetty joitakin heikkouksia. Tutkimuksessaan "Meta-Parametric Design" John E. Harding ja Paul Shepherd nostavat esiin algoritmipohjaisen mallintamisen monimutkaisuuden sen heikkoutena. (Harding & Shepherd, 2017)

Kohteiden suunnittelun kannalta hankesuunnitteluvaiheessa tehdään kriittisimmät päätökset hankkeen tulevaisuudesta. Parametrinen mallintamisen etuna pidetään sitä, että siten on helppoa kokeilla ja tutkia erilaisia vaihtoehtoja. Todellisuudessa parametrisesta mallista tulee helposti todella raskas ja aikaa vievä. Juuri tästä syystä alustavat suunnitelmat rakennuksista esitellään edelleen luonnosten ja jopa fyysisten pienoismallien muodossa, sen sijaan, että niistä luotaisiin tietomalli. (Harding & Shepherd, 2017)

Meta-parametrinen mallinnus on ratkaisu erityisesti hankesuunnitteluvaiheen luonnoksiin, jossa mallin tarkkuus ei ole välttämättä ratkaisevimmassa asemassa ja rakennuksen lopullinen muoto ei ole vielä varma. Tässä teknologiassa tietokone mallintaa parametrien avulla mallin ja tarjoaa erilaisia vaihtoehtoja annettujen reunaehtojen puitteissa. Eräs teknologiaa hyödyntävä ohjelma on Grasshopperin kanssa toimiva Embryo. (Harding & Shepherd, 2017)

Mitä kohdeyrityksen on tämän perusteella hyvä muistaa, on se, että siinä missä tietomallinnus oli kymmenisen vuotta sitten uusi juttu ja tulevaisuuden rakentamista, on se nyt arkipäivää rakennushankkeissa ja parametrinen mallinnus on suurelle joukolle alan ihmisiä tulevaisuuden rakentamista. Todennäköisesti kuitenkin asiantuntijoiden näkökulmasta uudet teknologiaharppaukset ovat jo tulossa, on seuraava suuri suuntaus sitten meta-parametrinen suunnittelu tai jotain aivan muuta. Myös kohdeyrityksessä on kehitystyötä suunniteltaessa ja kehitettäessä kiinnitettävä huomiota jatkuvaan kehitykseen ja alan

seurantaan. Kehitystyön on oltava jatkuvaa ja dynaamista, jotta vältetään jämähtäminen vanhentuneisiin järjestelmiin.

6. YHTEENVETO

Avoimet rajapinnat näyttelevät merkittävää roolia paitsi ohjelmistokehityksessä, myös tiedonvälityksessä ja jopa tiedon hallinnassa ja arkistoinnissa. Rajapintojen kehitys on kulkenut ensimmäisistä hotellien varauskirjoista lähes automaattiseksi osaksi sovelluksia ja palveluita. Avoimen rajapinnan merkitys esimerkiksi asiakastytyvääisyyteen on kiistaton ja sen merkitystä ei kannata väheksyä. Avoimet rajapinnat tuovat hyötyjensä myötä myös mahdollisia ongelmakohtia sekä avoimien rajapintojen käyttäjille, luojille ja omistajille.

Kaupallisten ohjelmistojen on harvoin tarkoituskaan olla valmiita, kun ne julkaistaan ja myydään käyttäjille. Tämä pätee myös Tekla Structures ohjelmistoon ja sen ymmärtäminen ja hyväksyminen on tärkeää, jotta yritys saa käyttöönsä kaiken ohjelmiston tarjoaman potentiaalin. Käyttäjien suuri määrä ja erilaiset tarpeet tekevät mahdottomaksi sen, että tuotettaisiin yksi kaikille sopiva valmis ohjelmistopaketti ja juuri tästä syystä Trimble tarjoaa paljon tukea ja opastusta avoimen rajapinnan käyttöön ja ohjelmiston muokkamiseen vastaamaan kunkin yrityksen erityisiä tarpeita. Tapoja hyödyntää Tekla Open API:a on lukuisia, joista osa helpompia toteuttaa vähäisemmillä resursseilla ja osa vaatii syvempää osaamista ja perehtymistä ohjelmointiin.

Kohdeyrityksen tarpeita ja nykytilannetta tutkittiin keskustelemalla kohdeyrityksen BIM managerin kanssa ja toteuttamalla kyselytutkimus kohdeyrityksen Tampereen rakennesuunnittelun yksikölle. Tutkimuksessa kävi selkeästi ilme, että käyttäjät ovat kiinnostuneita työnsä kehittämisestä ja Tekla Open API nähtiin mahdollisuutena parantaa työn tuottavuutta. Tutkimuksessa havaittiin myös muita työhön liittyviä kehityskohtia, kuten joidenkin kyselyyn vastanneiden tarve lisäopastukselle ohjelmiston parissa. Tutkimuksen vastauksissa annettiin myös uusia ideoita kehitystyöhön ja osoitettiin kiinnostusta uusien teknologioita kohtaan. Kyselytutkimus rajoittui tämän työn rajallisten resurssien takia vain yhden yksikön piiriin ja suuremmalla otannalla olisi voitu saada lisää kehitysideoita ja kattavampi kuva esimerkiksi työntekijöiden tämän hetkisestä osaamisesta ohjelmiston parissa, nyt puutteet ilmenivät vain yksittäisten vastaajien vastauksissa ja koska kysely suoritettiin nimettömänä, on mahdollista, että kyseinen yksittäinen vastaaja on esimerkiksi hyvin tuore työntekijä, jolla on perehdytys vielä kesken ja siksi osaaminen oli kyselyn toteuttamisajankohtana vajaata.

Huolimatta kyselyn mahdollisista vajavaisuuksista kohdeyritykselle annetaan neuvoja konkreettisen kehitystyön lisäksi myös perehdytysvaiheeseen ja miten ottaa huomioon

jatkuva kehitys koko yrityksessä. Jokainen työntekijä aina perehdytysvaiheesta alkaen nähdään voimavarana kehitettäessä uusia ja parempia työkaluja työn tueksi. Tulevaisuudessa kohdeyrityksen kannattaa ottaa huomioon myös tämän työn ulkopuolelle rajatut kaikki muutkin Tekla Structuresin käyttäjät kuin rakennesuunnittelijat.

Parametrinen mallinnus on yksi rakennusalan suurista trendeistä tällä hetkellä ja kohdeyritykselle suositellaan osaamisen syventämistä myös sen osalta osana Tekla Structuresin kehitystä, esimerkiksi ottamalla käyttöön Grasshopper lisäosa. Parametrinen mallintaminen voi osaltaan olla yksi ratkaisu alkuperäiseen tavoitteeseen, joka oli automatisoida toistuvia työtehtäviä ja vapauttaa suunnittelijoiden työaikaa muihin tehtäviin. Teknologia kehittyy jatkuvasti ja siksi pohdittaessa ja suunniteltaessa tietomallinnuksen tehostamista avoimen rajapinnan tai parametrisen mallintamisen avulla, on jatkuvasti syytä muistaa myös tarkkailla uusia kehityssuuntia ja ennakkoluulottomasti tartuttava uusiin mahdollisuuksiin.

LÄHTEET

ADN open - autodesk developer network. (2019). Retrieved from <https://www.autodesk.com/developer-network/open>

Aitamurto, T., & Lewis, S. C. (2013). Open innovation in digital journalism: Examining the impact of open APIs at four news organizations. *New Media & Society*, 15(2), 314-331. doi:10.1177/1461444812450682

Älykaupungit tehdään yhdessä. (2014). Retrieved from <https://6aika.fi/6aika-avoimia-ja-alykkaita-palveluja/>

Tekla tedds - luotettava ohjelmisto laskelmien tekemiseen. Arola, M. (Director). (2018, Oct 12.).[Video/DVD] Finland: Trimble. Retrieved from <https://www.tekla.com/fi/tietoa-meist%C3%A4/webinaarit/tekla-tedds-luotettava-ohjelmisto-laskelmien-tekemiseen>

Butterfield, A., & Gerard, E. N. (2018). Api. Retrieved from <http://www.oxfordreference.com/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-160>

Cater, M. (2013). A brief history of API-based web applications. Retrieved from <https://blog.smartbear.com/api-testing/a-brief-history-of-api-based-web-applications/>

Develop applications using tekla open API. (2019). Retrieved from https://teklastructures.support.tekla.com/2019/en/sys_tekla_open_api

Emmons, P. (2016). 4 benefits of an open API for customers, partners, and vendors. Retrieved from <https://www.dragonspears.com/blog/4-benefits-of-an-open-api-for-customers-partners-and-vendors>

Fisher, T. (2018). What is an EXE file? how to open, edit, & convert EXE files. Retrieved from <https://www.lifewire.com/exe-file-2622732?print>

Grillo, L., & Krebs, C. B. (2018). *Introduction*

Haikala, I., & Märijärvi, J. (2004). *Ohjelmistotuotanto* (10. uud. p. ed.). Helsinki: Talentum. Retrieved from <https://tut.finna.fi/Record/tutcat.174235>

Harding, J. E., & Shepherd, P. (2017). Meta-parametric design. *Design Studies*, 52, 73-95. doi://doi-org.libproxy.tuni.fi/10.1016/j.destud.2016.09.005

Jackson, O. (2018). PSD2 gives banks chance to evolve. *International Financial Law Review*, Retrieved from <https://libproxy.tuni.fi/login?url=https://search-proquest-com.libproxy.tuni.fi/docview/2007905178?accountid=14242>

Kartat ja paikkatieto. Retrieved from <https://www.maanmittauslaitos.fi/kartat-ja-paikka-tieto>

Kehittäjille. Retrieved from <https://vayla.fi/avoindata/kehittajille#.XKDoGpgzaUm>

- Kivekäs, O. (2012). Toimittajaloukku ja kuinka se vältetään. Retrieved from <http://otso-kivekas.fi/2012/07/toimittajaloukku-ja-kuinka-se-valtetaan/>
- Kivekäs, O. (2014, 16.6.). Mikä on avoin rajapinta? Retrieved from <http://otsokivekas.fi/2014/06/avoin-rajapinta/>
- lazar. (2018). What is an open API and why is it important? Retrieved from <https://www.techexpert.com/what-is-an-open-api-and-why-is-it-important/>
- Lehto, T. (2017). Pankit avaavat verkkopankkiensa rajapintoja - EU pakottaa kilpailuun ; *Tekniikka Ja Talous*, Retrieved from <https://www.tekniikkatalous.fi/tpaiva/pankit-avaavat-verkkopankkiensa-rajapintoja-eu-pakottaa-kilpailuun-6629520>
- Luukkainen, M., & Laine, H. (2012). *Luentomoniste kurssille: Ohjelmistotekniikan menetelmät*. Helsingin Yliopisto: Tietojenkäsittelytieteen laitos.
- Maanmittauslaitos.INSPIRE-velvoitteet. Retrieved from <http://www.paikkatietoalusta.fi/kunnille/inspire-velvoitteet>
- Oberg, R. J. (2002). *Introduction to C# using .NET*. Upper Saddle River, USA: Prentice Hall PTR.
- Paananen, J., & Granlund, K. (2005). *Tietotekniikan peruskirja* (6th ed.). Porvoo: Jyväskylä : Docendo.
- Penttinen, A., & Mäntynen, J. (2009). *Työhön perehdyttäminen ja opastus - ennakoivaa työsuojelua* (2nd ed.) Työturvallisuuskeskus TTK. Retrieved from <https://www.jytyliitto.fi/fi/jyty/materiaalipankki/Documents/Ty%C3%B6suhde/Ty%C3%B6el%C3%A4m%C3%A4n%20kehitt%C3%A4minen/Ty%C3%B6h%C3%B6n%20perehdytt%C3%A4minen%202009%20TTK.pdf>
- Poikola, A., Kivekäs, O., Kettunen, J., Polo, T., Laine, S., Aaltonen, J., . . . von Willebrand, M. (2014). Avoimen rajapinnan määritelmä. Retrieved from <http://avoinraja-pinta.fi/>
- Tanska, T., & Österlund, T. (Eds.). (2014). *Algoritmit puurakenteissa: Menetelmät, mahdollisuudet ja tuotanto* (1st ed.). Oulun Yliopisto, Arkkitehtuurin tiedekunta: DigWoodLab.
- Tekla developer center. (2019). Retrieved from <https://developer.tekla.com/>
- Tekla open API developer's guide* (2017).
- Thuray, A. (2013). How apis fuel innovation. Retrieved from <https://www.wired.com/insights/2013/12/how-apis-fuel-innovation/>
- Tossavainen, J. (2006). *Työhön perehdytys asiantuntijaorganisaatiossa*
- Trimble Solutions Corporation.Rakennesuunnittelu keskiössä. Retrieved from <https://www.tekla.com/fi/ratkaisut/rakennesuunnittelijat>

Trimble Solutions Corporation. (2018). Tekla open API training slides. Retrieved from <https://developer.tekla.com/documentation/tekla-open-api-training-slides>

Tryfonas, T. (2016). *Human aspects of information security, privacy, and trust*. Switzerland: Springer, Cham. doi:<https://doi-org.libproxy.tuni.fi/10.1007/978-3-319-39381-0>

Työhyvinvointi. Retrieved from <https://www.ttl.fi/tyoyhteiso/tyohyvinvointi/>

Työn kehittäminen. Retrieved from <https://www.ttl.fi/tyoyhteiso/tyon-kehittaminen/>

Väänänen, K. (2016). *TIE-04100 käyttäjäkokemuksen perusteet: Luentomoniste syksy 2016*. Tampere: Tampereen teknillinen yliopisto.

Varteva, A. (2016). *Avoimet ja yhtenevät rajapinnat*. (). Helsinki: 6Aika avoin data ja rajapinnat -hanke. Retrieved from https://6aika.fi/wp-content/uploads/2017/11/20171109_AvoimetYhtenevatRajapinnat_WEB.pdf

Varteva, A. (2017). Rajapinnoilla uutta liiketoimintaa. *Kauppalehti*, Retrieved from <https://blog.kauppalehti.fi/vieraskyna/rajapinnoilla-uutta-liiketoimintaa>

What are NURBS. (2019). Retrieved from <https://www.rhino3d.com/nurbs>

What is a DLL. (2018). Retrieved from <https://support.microsoft.com/en-us/help/815065/what-is-a-dll>

Yle api. Retrieved from <http://developer.yle.fi/>

Zumeran, R. (2017). The history of APIs and how they impact your future. Retrieved from <http://www.openlegacy.com/blog/the-history-of-apis-and-how-they-impact-your-future>

LIITE A: KYSELYTUTKIMUKSEN KYSYMYKSET

Taustakysymykset

1. Kauanko olet käyttänyt Tekla Structures -ohjelmistoa?
2. Kuinka usein tarvitset työssäsi Tekla Structures -ohjelmistoa?
 - monivalinta
3. Olen käyttänyt muita mallinnusohjelmia?
 - kyllä/ei
4. Tunnistatko käsitteen Open Api?
 - kyllä/ei

Ohjelmistoon liittyvät kysymykset

5. Anna arvosana Tekla Structures -ohjelmiston käytettävyydelle. 1 huonoin, 5 paras
 - monivalinta
6. Mikä tai millainen työkalu tehostaisi mielestäsi eniten työntekoasi?
 - vapaa sana
7. Mikä ärsyttää sinua eniten käyttäessäsi Tekla Structures -ohjelmistoa?
 - vapaa sana
8. Onko Tekla Structures-ohjelmistossa ominaisuuksia jotka turhauttavat sinua?
 - vapaa sana
9. Koetko, että käytät liikaa aikaa johonkin toistuvaan tehtävään, jonka automatisointi tehostaisi työtäsi merkittävästi? Jos kyllä, mihin?
 - vapaa sana
10. Mikä ärsyttää sinua eniten käyttäessäsi Tekla Structures -ohjelmistoa?
 - vapaa sana
11. Arvioi aika jonka kuukaudessa kuluu virhetilanteisiin tai ohjelmiston toimimattomuuteen.
 - vapaa sana

12. Olisitko itse kiinnostunut osallistumaan kehitystyöhön, mikäli se ei vaadi suurta tietoteknistä osaamista?

- vapaa sana

13. Vapaa sana